INTEREX COMPUTING MANAGEMENT SYMPOSIUM

ICMS '92

Proceedings

Sponsored by

INTEREX

The International Association of Hewlett-Packard Computer Users

MARCH 23-25, 1992 ■ ATLANTA, GEORGIA

INTEREX
COMPUTING
MANAGEMENT
SYMPOSIUM

INTEREX

the International Association of
Hewlett-Packard Computer Users

Proceedings

of the

1992 INTEREX
Computing Management Symposium

at

Atlanta, Georgia
March 23-25, 1992

Index by Paper Number

Index by Paper Number

Index by Paper Number

Index by Author

MANAGEMENT SKILLS FOR PERSONAL GROWTH

BUSINESS MANAGEMENT SKILLS

Index by Category

PRODUCTIVITY IMPROVEMENT THRU OFFICE AUTOMATION

DATA CENTER MANAGEMENT/NETWORKING

MANAGEMENT SOLUTIONS TO BUSINESS PROBLEMS

HP STRATEGY SESSIONS

ORGANIC MANAGEMENT

Keynote Address by

DANI WEINBERG

WEINBERG & WEINBERG
Rural Route 2, Box 32
Lincoln, Nebraska 68520
402/781-2542

ORGANIC MANAGEMENT

Our design of human systems in the workplace has not kept pace with our design of information systems. Our approach has been to manage by containment rather than by growth. In Euro-American culture, the work world is hierarchical, and this has been our chief model for organizing and managing. Various styles of linear management have treated people mechanistically and bureaucratically: we fit the person to the job. But there are other models by which people live and work productively.

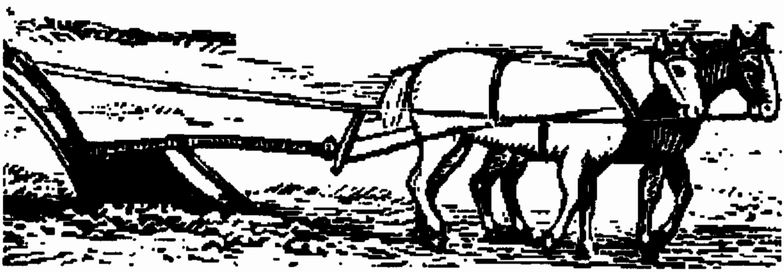
Organizations built on organic management are flexible and responsive. They encourage people to develop their personal, not just their positional, power. They support risk, uncertainty, and imperfection. They regard change as challenge, rather than crisis.

The remarkable history of human achievement on this planet depends on our three special abilities: to think, to learn, and to communicate. Organic managers are skilled at enabling and empowering these three uniquely human capacities in their employees and colleagues.

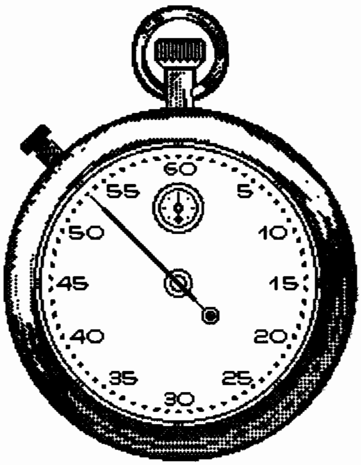
We have modelled the computer on the human being (the "electronic brain"). We must not make the grave mistake of modelling the human on the computer.

- What we mean by "management"
- Learning from history: management ideologies of the past
- Looking at people in the workplace in systems terms
- Anthropomorphic and mechanistic models of human organization
- Linear models: management by containment
- Organic models: management by growth

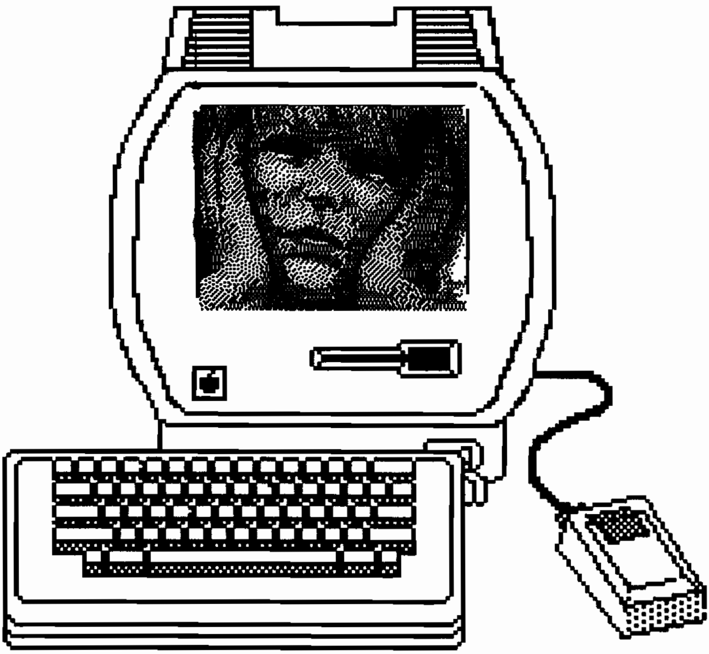
PEOPLE AS ANIMALS



PEOPLE AS MACHINES



PEOPLE AS COMPUTERS



MANAGEMENT IDEOLOGIES

- The Charismatic Entrepreneur (late 19th century)
  - The Career Manager (early 20th century)
    - Scientific Management and Rationality (early 20th century)
      - Industrial Human Relations (1930s)
        - Theory X and Theory Y (1950s)
          - Theory Z (1980s)
            - **What's next? (1990s)**

THE CHARISMATIC ENTREPRENEUR

"In his big transactions he seemed to act almost on impulse and intuition. He could never explain the mental processes by which he arrived at important decisions, though these decisions themselves were invariably sound. He seems to have had, as he himself frequently said, almost a seer-like faculty. He saw visions, and he believed in dreams and signs. The greatest practical genius of his time was a frequent attendant at spiritualistic seances; he cultivated personally the society of mediums, and in sickness he usually resorted to mental healers, mesmerists, and clairvoyants. Before making investments or embarking on his great railroad ventures, Vanderbilt visited spiritualists; we have one circumstantial picture of his summoning the wraith of Jim Fiske to advise him in stock operations."

- Burton J. Hendreck, The Age of Big Business, 1921
(quoted in Rosabeth Moss Kanter,
Men and Women of the Corporation, 1977)

SCIENTIFIC MANAGEMENT

<u>*Scientific Principles*</u>
Science, not rule of thumb.
Harmony, not discord.
Cooperation, not individualism.
Maximum output, not restricted output.
Development of each man to his greatest efficiency and prosperity.
- Frederick Taylor
(The Principles of Scientific Management, 1915)

1. Shift all responsibility for the organization of work from the worker to the manager; managers should do all the thinking relating to the planning and design of work, leaving the workers with the task of implementation.

2. Use scientific methods to determine the most efficient way of doing work; design the worker's task accordingly, specifying the precise way in which the work is to be done.

3. Select the best person to perform the job thus designed.

4. Train the worker to do the work efficiently.

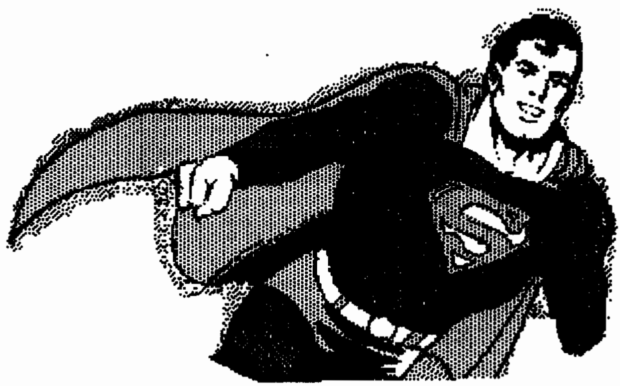
5. Monitor worker performance to ensure that appropriate work procedures are followed and that appropriate results are achieved."

- Frederick Taylor
(paraphrased by Marvin Weisbord,
Productive Workplaces, 1988)

INDUSTRIAL HUMAN RELATIONS

The leader "knows that the master of men has physical energies and skills and intellectual abilities, vision and integrity, and he knows that, above all, the leader must have emotional balance and control. The great leader is even-tempered when others rage, brave when others fear, calm when others are excited, self-controlled when others indulge."

- 1947 management manual
(quoted in Rosabeth Moss Kanter,
Men and Women of the Corporation, 1977)



THEORY X and THEORY Y

"Management's assumptions, said McGregor, determine management's behavior. McGregor suggested - Theory Y - that most people will take responsibility, care about their jobs, wish to grow and achieve, and, if given a chance, do excellent work. What stops them is managerial behavior based on Theory X, which assumes that most people are lazy, irresponsible, passive, and dependent, and must have their work broken into tiny pieces, tightly controlled, and supervised lest they make a mess of things. This was the theory that Taylor's scientific management had reinforced for decades." -Marvin Weisbond (Productive Workplaces, 1988)

THEORY Z

"Social organizations are incompatible with formality, distance, and contractualism. They proceed smoothly only with intimacy, subtlety, and trust. Z organizations...are most aptly described as clans in that they are intimate associations of people engaged in economic activity but tied together through a variety of bonds."

- William Ouchi (Theory Z, 1981)

THE GIFTS OF JUDGEMENT

- System in doing things
- Order in their possessions
- Living the planned life
- Capacity for sustained effort
- Exercising authority
- Enforcing rules
- Decisiveness, Settled opinions
- Acceptance of routine

THE GIFTS OF PERCEPTION

- Spontaneity
- Open-mindedness
- Appreciation of other viewpoints
- "Live and let live" attitude
- Curiosity, Zest for experience
- Adaptability

Ways of Perceiving the World

DEFINITION OF A RELATIONSHIP (How we perceive a pair)

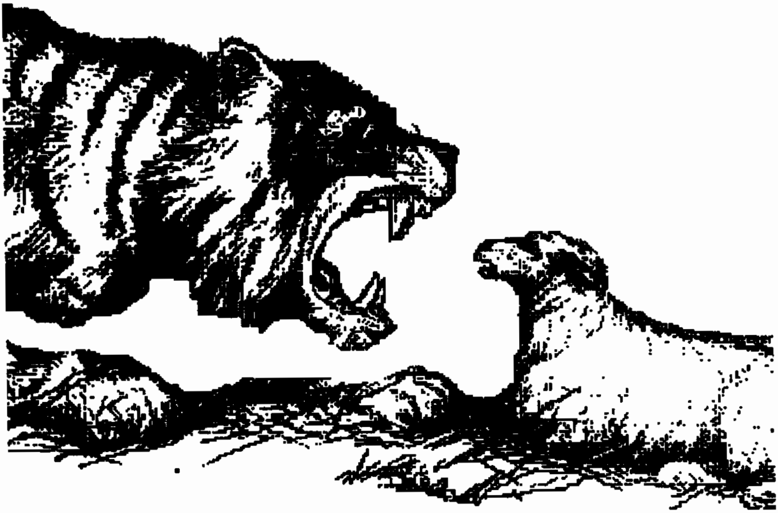
Hierarchical Model	Growth Model
People are of unequal value.	People are of equal value.
People dominate or submit to each other.	Relationships are between equals in value.
Roles and status are confused and blurred with identity.	Roles and status are distinct from identity.
Roles imply superiority and power, or minority status and powerlessness.	Roles imply a function in a specific relationship at a particular time.
The hierarchical view implies superiority and submissiveness.	Equality is manifested in: equality of persons, connection, interest and acceptance of samenesses and differences.
People have power over each other but feel isolation, fear, anger, resentment, isolation, and distrust.	People feel love, ownership of self, respect of others, freedom of expression, and validation.

DEFINING A PERSON

Hierarchical Model	Growth Model
People need to conform and obey "shoulds" for physical and emotional survival and acceptance.	Each person is unique and can define him- or herself from an inner source of strength and validation.
People are born with the potential to be evil.	People have an inborn spiritual base and sacredness, and they manifest a universal life force.
People are expected to think, feel, and act like each other, and to live up to external norms by competing, judging, comforting, and imitating.	Combining and respecting samenesses and differences, people delight in discovering themselves and others by cooperating, observing, and sharing.
People devalue or deny their feelings and differences.	People articulate their feelings and accept their differences.

Ways of Perceiving the World	
DEFINING AN EVENT	
Hierarchical Model	**Growth Model**
A causes B in a linear, cause-and-effect fashion.	Any event is the outcome of many variables and events. A = B + C + D . . . etc.
Only one right way exists to do something, and the dominant person knows what it is.	Many ways usually exist, and we can use our own criteria to choose an approach.
People deny their own experiences so as to accept the voice of authority.	People look beyond the obvious event to understand its context and its many contributing factors.
Thinking such as "That's the way it is" and "It's black and white" generates manipulation and shuts down originality and discovery.	Circular thinking and a systems approach (action-reaction-interaction) generate relevance, discovery, information, order, and connection.
ATTITUDES TOWARD CHANGE	
Hierarchical Model	**Growth Model**
Security requires maintaining the status quo.	Security grows out of confidence in the process of change and growth.
People view change as undesirable and abnormal. They therefore reject and resist it.	People view change as ongoing, essential, and inevitable. They therefore welcome and expect it.
The familiar is more valued than the comfortable, even if the price is painful.	People view discomfort or pain as a signal for change.
People fear the unknown.	People take risks and opportunities to move into the unknown.
People judge changes as being right or wrong.	People delight in discovering new choices and resources.
People feel fear and anxiety when they face the prospect of change.	People feel excitement, connectedness, and love when they encounter the prospect of change.

LINEAR MODEL



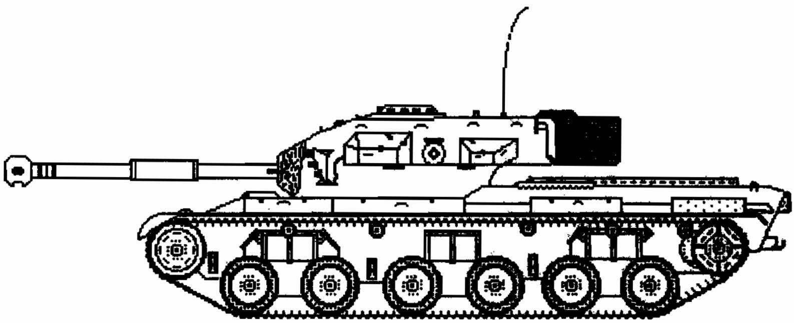
"...competition dampens creativity. This is partly because the pressure to outdo someone else tends to make us conservative. We do not want to risk anything that could endager our victory....If winning is the goal, one naturally tries to avoid doing anything that could jeopardize it. If I want to get the highest grades in class, I will not be likely to challenge the teacher's version of whatever topic is being covered. After a while, I may even cease to think critically altogether....If people tend to 'go along to get along,' there is even more incentive to go along when the goal is to be number one. In the office or factory where co-workers are rivals, beating out the next person for a promotion means pleasing the boss. Competition acts to extinguish the Promethean fire of rebellion....Creativity is anticonformist at its core; it is nothing if not a process of idiosyncratic thinking and risk-taking. Competition inhibits this process."

- Alfie Kohn (No Contest, 1986)

MECHANISTIC MODELS

"In Max Weber's work, "we find the first comprehensive definition of bureaucracy as a form of organization that emphasizes precision, speed, clarity, regularity, reliability, and efficiency achieved through the creation of a fixed division of tasks, hierarchical supervision, and detailed rules and regulations."

- Gareth Morgan (Images of Organization, 1986)



ORGANIC MODEL

"The relationships that succeed can be described simply. Within the organization, people work as a team, are open with each other, demonstrate their presence as individuals, and show their respect and esteem for each other. They treat each other as unique, are aware of and build on their sameness, and grow and learn from their differences. They model the behaviors and values they wish to teach their subordinates. Conflict resolution (dealing with differentness) becomes one vehicle by which new growth occurs. New members learn how to do this by watching the people around them. To practice what they preach, to model in this way, managers need to have developed a high degree of self-esteem."

- Virginia Satir (adapted from The New Peoplemaking, 1988)

"I wonder what would happen if suddenly during one night, all five billion persons in the world learned the essentials of congruent living:

To communicate clearly
To cooperate rather than compete
To empower rather than subjugate
To enhance individual uniqueness rather than categorize
To use authority to guide and accomplish 'what fits' rather than force compliance through the tyranny of power
To love, value, and respect themselves fully
To be personally and socially responsible
To use problems as challenges and opportunities for creative solutions

I think we would wake up in a very different world, a world in which peace is possible."

- Virginia Satir (The New Peoplemaking, 1988)

KEY IDEAS FOR THE 90s

Block, Peter

1987 The Empowered Manager: Positive Political Skills at Work. San Francisco: Jossey-Bass.

Clearly and incisively written, this book makes a strong statement in support of participative (organic), as opposed to bureaucratic (linear), management styles. Block outlines a vision of how organizations can be and explains ways for people in management positions to make them so.

DePree, Max

1989 Leadership is an Art. New York: Doubleday.

DePree, the CEO of Herman Miller, Inc., in this gem-like and quite personal book, gives us a description of organic leadership, defining it as "liberating people to do what is required of them in the most effective and humane way possible."

Kanter, Rosabeth Moss

1989 When Giants Learn to Dance. New York: Simon & Schuster.

This book offers a rich and extensive argument for approaching management in a perceptive style. Kanter makes the case for moving away from "segmented" and towards "integrative" organizations.

Keirsey, David and Marilyn Bates

1984 Please Understand Me: Character & Temperament Types. Del Mar, Ca: Gnosology Books.

This is the best overall introduction to the ideas of Jung and Myers on how people differ in their preferences for Perceiving and Judging and in their attitudes towards the world.

Kohn, Alfie

1986 No Contest: The Case Against Competition. Boston: Houghton Mifflin.

Kohn argues persuasively that competition – "gaining success by making others fail" – is destructive and unproductive. He effectively refutes our culture's myths about the presumed benefits of competition, basing his arguments on extensive scientific research on human behavior. This book has helped many executives move towards the organic model of management.

Morgan, Gareth

1986 Images of Organization. Beverly Hills: Sage.

A very readable overview of the history of organizational forms and management styles in the U.S. This book gives us perspective on how we got to where we are today and helps us see possible futures.

Myers, Isabel Briggs

1980 Gifts Differing. Palo Alto, Ca: Consulting Psychologists Press.

Myers published this book shortly before her death. It is her own definitive statement on type, containing excellent discussions of the four personality dimensions and a wealth of data on their distribution in different occupational groups.

Ouchi, William G.

1981 Theory Z: How American Business Can Meet the Japanese Challenge. New York: Addison-Wesley.

This book comes out of the era of "Japan paranoia" in U.S. business culture. Much of Ouchi's discussion was misunderstood to mean that we could import Japanese management practices without any cultural translation. The book, therefore, generated a spate of out-of-context "reforms" such as quality circles. We can now reread it for its insights into the organic model.

Satir, Virginia

1988 The New Peoplemaking. Mountain View, CA: Science and Behavior Books.

This is an updated version of the classic book that made Satir famous. It was published just before her death in 1988 and presents her powerful ideas about congruent communication and the process of change. For more information about her books, workshops, and videotapes, contact:

Avanta Network
139 Forest Avenue
Palo Alto, CA 94301

Vaill, Peter B.

1989 Managing as a Performing Art. San Francisco: Jossey-Bass.

This thoughtful and insightful book helps us rethink traditional management strategies in an age of intense change. Vaill advocates "a mentality that is friendly to paradox" (in Chapter 5) – essentially, a Perceiving (in Myers-Briggs terms) approach to management to counterbalance our heavily Judging organizations.

Weinberg, Gerald M.

1986 Becoming a Technical Leader: An Organic Problem-Solving Approach. (Foreword by Ken Orr) New York: Dorset House.

This book is a personalized guide to developing the qualities that make a successful leader. It identifies which leadership skills are most effective in a technical environment and explains why technical people have characteristic trouble in making the transition to a leadership role. Weinberg makes a strong case for organic thinking in the development of technical organizations.

DANI WEINBERG

Dani Weinberg has consulted, published, taught, and lectured extensively on organizational cultures - how they work and how they change. She is especially interested in empowering people to make their organizations more satisfying and productive.

As a principal of Weinberg & Weinberg, an international consulting firm that focuses on the human side of technology, she is involved in customized consulting and training in such areas as organizational development, technology transfer, leadership, problem solving, group process, gender issues, and management of differences.

Dani and Jerry Weinberg's world-famous *Problem Solving Leadership* workshop, has been evolving for two decades. Hundreds of women and men from a wide variety of countries and industries have attended. Participants have come from Pacific Bell, U.S. West, Aldus, Microsoft, Arthur Young, L.L. Bean, Philips Research Laboratory (Netherlands), Prudential/Tesseract, The Country Companies, Amdahl, Ericsson Information Systems (Sweden), McDonnell Douglas, MCC, Tektronix, Digital Equipment, IBM, Dylex (Canada), Deseret Mutual, St. Paul Fire and Marine, U.S. Bureau of Engraving and Printing, and Port of Portland.

In addition, the *Congruent Leadership Change-Shop* was created by Weinberg & Weinberg with Jean McLendon to enhance the personal and professional effectiveness of people in organizations. Designed for maximum and immediate take-home applicability, the Change-shop's primary objective is to enable participants to lead their organizations through change towards creating more fully human workplaces.

Dani Weinberg is also Adjunct Professor of Anthropology at the University of Nebraska-Lincoln, where she has done research on a wide variety of subjects, ranging from Swiss Alpine agricultural societies to the cultures of corporate America.

Among her publications are *General Principles of Systems Design* (formerly titled *On the Design of Stable Systems*), with G.M. Weinberg, and *The User Interface* column, with G.M. Weinberg (Journal of Information Systems Management).

MAINTAINING A QUALITY STAFF STARTS DURING THE INTERVIEW

JEFF ODOM

BAHLSEN INC.
ONE QUALITY LANE
CARY, NC 27513
(919)677-3227

"One job belongs to the leader alone; and that is making sure all the parts and all the people work together."[1]

You can have the best parts in the world - yet the truck you build may not run well, if at all. You may have the best flour, sugar, spices, oils, etc., but the cookies you bake may not taste good and may be hard as a brick. How is it people can shop at the same supermarket, buy the same ingredients and use the same recipe yet the final product be as different as the east is from the west? The functioning of people, their inter-relationships, skills, motivation and quality consciousness bring the ingredients together and produce quality goods and services.

We are responsible for blending associates to create a quality team. "In a system everything has to work together harmoniously. The parts and the people have to be optimized to work with each other toward a common goal."[2] You must be sure to fit the right person into the right position by viewing the whole employment setting and pay particular attention to intangibles. Success will always come down to intangibles of fit, personality and relationships.

Unfortunately, key organizational ingredients may be missing if your company does not subscribe to philosophies that create fertile environments for associates to develop. Even so, you must do what you can to create a divisional, departmental or workgroup environment which will promote such associate growth. Some of these concepts include full employment without fear, training, quality, mobility organizationally and so forth. People want to perform in a manner that provides them the satisfaction of a job well done. The work place must allow associates to perform to the best of their abilities without fear of reprisal and be fully trained to perform duties as assigned.

Quality work relationships, products and services depend upon individuals filling appropriate roles on a team. Our world is full of examples; such as consistent top flight college basketball programs built on quality team defense. Another favorite example is of an orchestra. In an orchestra, each member is a qualified and talented musician, but to perform as a team each must fill their role at the appropriate place and time under the conductor's leadership (coordination). If each member of the orchestra were to try and highlight their individual skills, the harmony which creates the beauty of the music is lost.

"Quality is pride of workmanship or joy of work. By allowing and even urging workers to experience intrinsic rewards that come from doing something well, using their innate and acquired abilities, productivity improves, quality improves and customer satisfaction improves."[3]

Our responsibility is to place the right individual in the right position. Further, create an environment that allows for pride of workmanship and personal development. By doing this, maintaining quality staff will happen naturally. This is not setting high standards and sticking to them often demanding the impossible. Nor do we have to reward excellence and punish so called mediocrity. We must empower the individual with dignity, knowledge and skills so that they may contribute to our team.

So, what is your environment? Corporate? Divisional? Departmental or Work Group? Obviously we hope, all the way up, it is an environment of individual empowerment, quality, cooperation and teamwork. It just makes sense that this is the environment of success. Varying combinations at different levels of your organization have to be factored in. Our hope is your organization will come to terms with this because truly this is what we all want as candidates for positions.

Understand that this organizational environment or system is key to associate success. The most important factor in any hiring decision and success of a candidate, no matter what the degree of the appropriate organizational environment, is the personality fit of candidate/associate and immediate supervisor. Talent, technical skills, motivation, education and so forth are important qualities but the ultimate success lies in the relationship between supervisor and subordinate.

We may say that the keys to successful hiring are:

1. Associate and supervisor fit
2. Organizational philosophies
3. Skill, experience, education fit
4. Team fit

If you notice, we as management, control or highly influence points 2, 3 and 4. We, to a varying degree, influence corporate philosophies and should be responsible for our own team development. As for point three, we select the appropriate skills or they are obtainable by providing education and training to associates. All this leaves us with two dilemmas of making sure we truly have the appropriate skill set and more importantly how to determine supervisor/associate fit.

May I suggest a methodology that has worked very successfully for me and was derived in large part by the teachings of Dr. Kurt Einstein noted behaviorist. Dr. Einstein taught his methodology in a seminar all over the world called "Picking Winners." The following is how I use his method incorporating processes influenced by other behaviorists as may be noted.

The basic process is:

1. Define the position to be filled

2. Determine a candidate's
   a. must haves
   b. preferred to haves

3. Advertise or network for candidates

4. Preinterview planning

5. Interview

6. Second interview if necessary - or is it always necessary?

Defining the position then developing the must and preferred to haves is critical to the selection process. The definition determines who will be interviewed. We don't want to exclude a qualified candidate and we also cannot spend precious time interviewing candidates that do not fit our needs. Further, once hired, this definition is given to the associate as a clear representation of what they will be held accountable for.

To define the position we must determine the following:

1. What will the person do?

2. What will their responsibilities be?

3. What decisions will the individual be allowed to make in 3 months, 6 months and a year. This is an important process in developing associate/supervisor trust. Robert Tannehill wrote of the importance of this point as it pertains to managers: "For an organization to give managers freedom to make mistakes, to grow and to learn, there must be a fairly high level of trust and confidence among the managers one with another, and especially on the part of top management. When a mistake is made, it is often the practice of management to scurry about to try to find out who is to 'blame' for the error."[4]

   I always tell associates immediately you have the right to make mistakes, your obligation is to communicate to the appropriate individual(s) as soon as you realize the mistake. By setting out clearly when decisions are the associate's responsibility, there is comfort and greater ease in the decision process.

4. What technical skills and education does the position need?

Having done the necessary to understand our position, received and reviewed resumes and set appointments with qualified candidates, we move to the critical process of interviewing the candidates. The interview process is an evaluation process - there is a mutual decision being reached. Both the candidate and the interviewer (organization) have to reach a quality decision as to fit. Dr. Einstein taught in his seminars that there are two reasons why personnel selection decisions fail:

1. Failure to evaluate fit of selected person.

2. Failure of selected person to evaluate fit (confusing specifications or poor understanding of specifications).

If we understand what causes personnel decisions to fail, we then can define the goal of the selection process. Simply, our goal is to avoid the failure issues by increasing our ability to make predictive selection decisions. To do this we want to evaluate:

1. The capability of the person. What a person can do leads to performance. What a person can explain to us in detail they can do.

2. The personality of the person. What a person is like leads to behavior.

Knowing these two points, we will be able to predict how a person will perform and act. Prior to the interview, review the candidate's qualifications to determine strengths, weaknesses and fit to your criteria. Determine appropriate questions that will validate the strengths or true attributes and the extent of perceived weaknesses. Remember we all have strengths and weaknesses so our objective is to make an informed decision as to the candidates fit given both. We are not looking for someone who has no weaknesses.

To make this evaluation, we want to conduct the interview as follows:

1. Set a comfortable environment overtly free from interruptions and distractions.

2. Ask for agreement in the process. Ask may I take notes? Do you mind if we sit here?

3. Explain that this is a mutual decision process and the penalty for error is high for both parties. Further, both parties are responsible and accountable for the decision.

4. Direct the interview in a fashion that you ask all of your questions first then allow the candidate to ask any and all questions. Have questions flowing one way.

5. Ask open-ended questions. Avoid yes and no responses.

6. Ask why, why, why! Probe pat and standard response by exposing the reasoning behind them.

7. Ask revealing questions as described below.

One of the problems in interviewing is determining what questions will stimulate responses that will most add value to your decision process. What is proposed here is that you ask open-ended questions to which the response will tell you about the true capabilities and personality of the candidate. Below is a list of example questions from Dr. Einstein's seminar as well as those other successful interviewers in our organization have used.

1. What would you describe as the best attributes of your current supervisor?

2. Tell me about two serious interpersonal relationship problems you've had on your job(s).

3. What are your likes and dislikes? These are from all phases of life; personal, emotional and career.

4. Give some examples of situations where you have been criticized. How did you react and why?

5. Under what conditions do you feel you learn the best?

6. To what extent do you feel that your college grades accurately reflect your ability?

7. What do you feel are your three most significant accomplishments? In detail describe how one of them was achieved.

8. Describe the perfect boss.

9. Under what conditions would you consider yourself as "having arrived" in your personal and professional life?

10. How do you resolve family differences?

These questions give you a flavor for the types of things to ask to discover who the person is and what their capabilities are. For technical aspects, you would key in on accomplishments and the detailed description of what took place. For supervisors, look for questions that reveal personal behavior skills (so called soft behavior). In all questions, continue to seek answers which will allow you to make the best predictive selection decision you can.

The complexity of interpersonal relationships is vast. Our probability of success as organizations increases as we select the best candidate for a given position. It is important that we continue the development of all associates once hired. Give your staff an opportunity to take responsibility of their own future by empowering them with training and security.

---

1. Rafael Aquayo, Dr. Deming the American Who Taught the Japanese About Quality (Carol Publishing Group, 1990), p. 180.

2. Ibid.

3. Ibid, p. 45.

4. Robert E. Tannehill, Motivation and Management Development (Auerbach Publishers, 1970), p. 81.

ARE YOU MY PRIORITY?

JEFF ODOM

BAHLSEN INC.
ONE QUALITY LANE
CARY, NC 27513
(919)677-3227

It is most probable that all of us have participated in a minimum of one time management course by now. It is also highly likely that we are using more than one of those modern "time saving" devices (computers, faxs, cellular phones, etc.). There is also a good chance we actually use our "time saving" devices to do more as opposed to saving time. We could actually say we probably are better educated in time use, do more in the time we have and ultimately, we remain a bundle of frayed nerves. Instead of having the four day work week promised a decade ago, we now are stressed out by the demands of intense multi-tasking.

Often we become unsettled if we do not have dozens of balls in the air simultaneously. We are stressed if we do, stressed if we don't. Where do we turn? Of course, we turn to our training! We change our "bad" behavior into "good" behavior. To do this we remember that they told us true change will not take place before three weeks, we will not change if we do not start within 24 hours of learning how, we must remember this is good for us (no pain, no gain?), know your priorities, everyday study the list of time wasters, absolutely use the miracle tools the trainer so graciously let us in on, and on and on and..........

In reviewing and participating in so called time management courses, I have found everything from outright product pitches to positive attitude promotions. One course I attended covered the following subjects all in a six hour ordeal:

- Personality profile
- Motivating others
- Concentration and alertness levels

- Task juggling
- Time management assessment
- Time management tools
- Delegation
- Procrastination
- Team building
- Positive attitude and words
- Lifestyle
- Job stress
- Assertiveness evaluation and training
- Communication
- Time waster tips

It seems the developer of the course needed some time management training to realize you can not adequately cover and integrate all of the above concepts in so short a time. Granted we need the whole of these concepts if not a few more, to effectively manage our time; but let us please have digestible portions! Further, when it is all said and done, how will I get done what I need done?

What is missing is the engine that drives the other time management techniques and tools. We need help setting priorities. What's important right now! I have even attended courses whose title included words such as managing or setting priorities, yet spent virtually no time on the subject. How could they with all the above listed topics to cover? What was their priority? Maybe it was as simple as duping us into one more time management course.

Actually, we are not putting down time management courses. Implementing the tools and techniques generally taught in them will clearly improve your plight. It was during a particular session whose title connotated help with priority setting but whose subject actually was a little time management tools and a lot of people management that my mind began to wander. How can we characterize and manage priorities? I began an animated daydream that paralleled P.D. Eastman's wonderful children's book Are You My Mother?. If you recall, the story is of a little bird who hatched while the mother was off looking for food. The hatchling wonders where it's mother is and begins searching, meeting other animals and some machinery in the search inquiring "are you my mother?" Ultimately the "Snort" (steam shovel) places the young bird back in the nest just in time for the mother's arrival with dinner.

If we could, let us take a few moments to share this adventure in recognizing the forces setting our priorities. But before we begin, take a moment to write on a piece of paper in priority order the 6 things you would do with your remaining time (including revenue generation) if you found out today that you would live in 100% health but die overnight exactly 6 months from now. What are the things you would want to do? Now set the paper aside and let us ask of the following tasks or personality traits found in others and yourself, are you my priority?

- ☺ The Bear - Threatening and powerful, successful at intimidating us.

- ☺ The Snake - Sneaky and cool, winding into our time.

- ☺ The Otter - Full of life and mischief, never a care for now, we will get to it later.

- ☺ The Lion - Loyal and powerful, seeing the world in the light of loyalty. Dignity and grace further adds to their influence.

- ☺ The Rhino - A very territorial beast who defends its space with charging and snorting. Not like a bully this one will defend to the death it's turf.

- ☺ The Kitten - So sweet and cute to "purrrrfection" ease you away from your sense of direction.

- ☺ The Rabbit - This one is cute but so jittery to boot. You begin to be attracted out of compassion for a creature such as this.

ARE YOU MY PRIORITY?

- ☺ The Elephant - The most commanding of all of our animals yet regal and unassuming, content with steering the course.

- ☺ The Steam Roller - Compacting and crushing all that falls in front of its path.

- ☺ The Fire Truck - Everything is a crisis, for that's it's job. Sometimes we wonder who's setting those fires?

Are these our priorities? Well yes in part, but the list you made a few moments ago are your true life priorities and all of the other things in your life must first be matched up to this list then set as a priority based on your own time management technique (A, B, C's or I, II, III's, etc.). Use this list as your priority targets. From your target you can establish goals which will help identify your day-do-day and moment-to-moment priorities. Granted, the many influences of people and work tend to draw you this way and that, but keep it all in perspective of your larger whole life priorities and you will stay on track with your goals.

Use the tools that work best for you in maintaining calendars, to do's, delegation and so forth. Remember the difference between important and urgent. Through people skills and assertiveness training, learn to manage the influence on your day-to-day priorities. Evaluate your priorities throughout the day, work in bite size pieces and avoid procrastination. These are all important to managing our time. The key though, is to thoughtfully evaluate what is important in the way you spend your time. Stop and evaluate "Are You My Priority?" based on a sound and healthy knowledge of what your true priorities in life are.

ICMS '92
March 23-25
Atlanta, Georgia

Paper Number: 1003

User, Techie or Both?

Tena L. MacDonald
Indianapolis Water Company
1220 Waterway Blvd.
Indianapolis, IN 46202
317-263-6380

User, Techie or Both?

This session is for the MIS individual that feels a barrier exists when they or their associates communicate with "users" and would like to break down that wall of confusion and misinterpretation.

Some familiar examples:

Terminology	User Perspective
Network	CBS, NBC or ABC?
JCL	Is JCL a new bomb? It blew up last night.
	Programmer Perspective
Credit AP for Inventory	Why does Alice Pica deserve recognition for Inventory?

Both parties become insecure when trying to speak the others' language. If there is not some attempt to talk the same language, people end up committing to procedures, projects and steps they do not really understand.

Additional illustrations will be given along with solutions and alternatives that are taking place within business today to try and circumvent this communication problem.

Scenarios

User calls:	"I asked for a report last night but it was not on my printer this morning."
Programmer:	"The JCL blew up - I'll restream it and you should have the report in about an hour."
User thoughts:	JCL? What is that? Restream? What was the problem?

Alternative response by programmer: "There was a problem with the request, not from your entry but something on our end. I will ask for the report again, after I correct the problem, and you should have it in about an hour."

MIS Strategy:	We can network PC's together and spool all reports to LP.
Reworded:	We can link personal computers together and send all reports to the main printer.

User, Techie or Both?

User comment: This entry is not working properly. It should debit inventory and credit AP.

Programmer thought: Which record is that? Does debit mean add or subtract? Should we be increasing or decreasing the AP (Accounts Payable) account?

User Question: "Should I go ahead and post the entry?"

Programmer: "Sure? Where are you going to post it?" (bulletin board)

User response: "To the General Ledger, of course!"

Programmer: "Oh! You are going to apply it to the database!"

I attended a committee meeting for a local school system. The MIS group was discussing the evaluation and selection of financial software. They were discussing the system requirements and how each vendor measured up, then all at once a school board representative spoke up: He said "I do not understand what all of you are talking about. I just want a list of requirements that I can take to the school board in order to get their support and financial backing for this expenditure." The MIS people said they had that information. For example, Purchase orders will be added on-line, all satellite schools will be on-line to the central computer, etc. Non-technical people, who are not familiar with automation, need to hear benefits, not requirements. A benefit means taking a current situation and stating how that will be eliminated or improved. For example, there is a current time lag between requesting a purchase order and receiving against the purchase order. Accounting might send a request to purchasing for a purchase order, Purchasing will send a form back to Accounting, Accounting will fill it out and have purchasing place the order, paperwork then has to be found in the files when the invoice is received. Benefit of new system: A purchase order will be added to the system immediately so the items will be able to be received the same day so the general ledger will immediately reflect our liability. Also this will reduce the amount of mail that is sent back and forth between departments.

These examples portray the breakdown in communication and uncomfortable feeling felt by each group as they try to understand what the other is saying.

Presumed Solutions

Each group has their recommended solution. MIS feels the user department must learn technical terminology, on their own or by asking questions (admitting their lack of knowledge), if they intend to understand anything MIS says. Many users are doing just that by taking classes or becoming more computer literate as PC's are more popular, accessible and practical for word processing, spreadsheets and, of course, games! It is funny when they try to use technical language but do not really know what they are saying. For instance: "Print the spooler." The user as well as the MIS-type knows they mean print the report.

On the other hand, users feel MIS should learn their jargon and understand business dealings so they do not have to explain basic business concepts to them every time a new MIS person works on a project for them. MIS always wants to free up disc space, purge old data, etc. They do not understand there are federal requirements for keeping old data for specified retention periods. Departments do not keep this data because they may look at it some day - there are federal regulations regarding certain types of information.

Best Solution

Obviously, the best solution is to work together to learn some of each others language. Build an environment that encourages "Ask if you do not know or understand the terminology" and refrain from embarrassing or having a condescending attitude toward those that do ask. Each needs to gain exposure to the other in order to be more effective in their work environment. If one area shows interest in truly understanding the business of the other - you will see a more positive attitude of all parties involved.

Theory vs. Reality

I know some of you are saying, "That sounds good in theory, but does it really work?"

Obviously, this exchange of terminology can take place but if the audience is not receptive you continue to have two sides (Them & Us). As with any business - everyone needs to work together, rather than be divided. Being divided not only affects business productivity, it affects the work environment and opportunity for all to enjoy where they spend at least 40 hours, 52 weeks a year.

Suggested Actions

Users can become informed by providing them with a dictionary of common terminology used by the MIS staff in your company. (A sample, to get you started, can be found in Appendix A). Do not expect them to remember everything after the first glance - reinforce concepts as they come up in conversation.

The MIS department should have a library of basic Accounting, Purchasing, Marketing, etc. books for their reference.

MIS plays an integral part in supporting a company by processing and presenting data for the decision makers. In order to contribute to the effectiveness and value of this data, MIS associates must have a grasp of the business as well as understand each departments terminology. They need to know the answers to the following questions:

1. What is the mission of the company?
2. What is the philosophy of the company?
   The Best; We try harder; Status quo; Struggling to get ahead

3. What is the primary focus of the company?
   Better prices from vendors; More customers; Improve product; Environmental concerns; Shareholder opinion; Quality service; Good Corporate citizen
4. Who are the suppliers?
5. Who are the customers?
6. What is the business flow?

MIS is the only department that can see the complete use of data by various departments. They can contribute greatly to the effective use and presentation of the information when they comprehend the goals and motives of those needing the data.

There needs to be a silent leader of this philosophy to help MIS learn English as a second language. It only takes one person. Stress classes and seminars that will assist in giving the MIS staff an understanding of business areas.

-MIS department meetings should discuss business of the company and the flow of data between departments, rather than logic in a program. Programmers can read and figure out code, it is discussing the companies business that allows them to be smarter as they design systems and make decisions along the way.

Some sort of incentive may be necessary for those techies that are not already interested in business and communication with users. Their idea is "The less they(users) know, the better off I am (MIS)". Talk about a dead-end thought process. I am not saying they need to know everything but working in a shell is not productive when employed by a company.

-Begin a penny jar or award system: For every techie statement said to a user - they put a penny in the jar. A techie that rephrases a sentence for a user takes a penny from the jar or gets some other reward.

-Have the users vote on the most "user friendly" MISer.

New Approach

Learning the jargon of the user departments and the business of the company does make you more marketable. User departments are now looking for people that know MIS and English to join them and become their liaison to the MIS department. What a terrific need to be able to fulfill!

The more people that move from MIS to other areas in the company, the weaker the central MIS department will become. If they want to maintain any sort of unity they must adhere to this advice. Any analyst or support function that interfaces with other departments, must be able to speak in laymen language and understand the goals of the company as well as the goal of the MIS department in treating internal associates as customers.

Two large corporations that are moving their MIS staff into their business areas are UNUM Life Insurance Corporation of Portland, Maine and Dial Corporation in Phoenix, Arizona. UNUM, rather than using the term "decentralization", prefer to say they are "integrating system development people into the business arena". Dial Corporation is focusing on business work units that consist of user department staff as well as MIS individuals. They feel the level of distrust between the groups will diminish, system modifications will be more timely and new development projects will be more applicable.

Are you a User, Techie or Both?

Good luck to you as you increase your vocabulary and instill that same desire in others!

Appendix A - Techie Talk

CPU - Central Processing Unit (Box that contains the brains)

LAN - Local Area Network. Connection between machines (PC's or mainframes)

GIGO - Garbage In, Garbage Out. This refers to data entry and relects the need for accuracy in order to benefit from the data at a later date.

DOS - Disc Operating System. This is the main program that tells the computer what to do and how to do it. (Brain)

C Drive - On a personal computer this is normally your primary storage space (the default). Alternative storage spaces for data are diskettes that you insert and remove from your floppy disc drive. (I know, what is a floppy disc drive?!)

WYSIWYG - What You See Is What You Get. You can view results on your terminal before printing.

Hp - Hewlett Packard

Network - Linking machines together so they can share the same data.

GUI - Graphical User Interface. Ability to show pictures on the screen, rather than words, for users to select.

Bug - Statements in a program that are not working as expected.

Byte/Positions/Characters
Number of spaces that represent the number of letters in a word or number of digits needed to print/display a number.

PC - Personal Computer. Can do work on its own without being connected to another machine.

Bibliography

"Bridging the Gap between I/S and Users" IBM Directions, Summer/Fall 1991, Vol 5, No 2, pp. 20-23.

Robinson, Michael A. "Decentralize and Outsource: Dial's Approach to MIS Improvement" Management Accounting, Sept. 1991, pp 27-31

THE TECHNICAL TO MANAGER TRANSITION

A Formula for the 90's

Ronald E.Stanton, CDP
RES Software Products
P.O. Box 66
Fairport, New York 14450-0066
United States of America

INTRODUCTION

The close of the 1980's marked an end to continuous growth and profitability for many organizations. Increased pressures on the bottom-line forced companies to trim overhead and waste. When the inefficiencies disappeared, the next areas to be scrutinized were the middle and lower management ranks.

As the 1990's began, 'lean and mean' was the new motto for most MIS organizations. Doing more with less was commonplace. Our estimates were requested to be lower and demands on our time were much higher. The role of MIS managers may have changed more since 1990 than in the previous decade. These changes require new managers to seek a formula for success. Without fine-tuning our perceptions of management's role, success in the 1990's may be a bitter-sweet prize.

A VIEW OF THE TECHNICAL ROLE

A primary key to success in the technical support role was intense training. The ability to perform was based on a well-defined skillset and intimate knowledge of technical reference materials. Constant use normally increased our efficiency and performance ability. To learn more, we only needed to dig into a manual and try some new technique. No one would even know that we were doing it. If the new technique didn't work, we could simply fall back on proven methods.

Responsibilities revolved around fixed assignments that were at least partially defined. Delivery dates were usually pre-defined and sometimes, arbitrary. The scope of our task assignments was often narrow, rarely giving us an opportunity to view the total project. Attention to details was an absolute must, especially for sensitive systems. We usually worked with a limited set of tasks and completed them based on our knowledge and experience. In summary, almost all of our work was completed with respect to specific limitations and guidelines.

However we attempt to interpret the technical role, involvement is clearly restricted to the delivery system. The definition/planning processes normally happened prior to the technician's assignments. In most organizations, knowledge of prior development stages is minimal. Technical support personnel rarely have opportunities to develop skills that are outside the delivery system arena.

A VIEW OF THE MANAGEMENT ROLE

In order to effectively review the role of MIS management, it may be beneficial to briefly cover the evolution of MIS in corporate America.

In the early days of automation, the role of data processors was viewed as 'computerization of manual-intensive tasks'. User support/customer service concepts really didn't appear until the 1970's for many companies. As the 1980's passed, productivity and quality assurance were terms whose emphasis became focused on the MIS organization.

An observation by Arlene Duggan(VP, DYNAX Personnel Services of Jericho, New York), '...the computer system has assumed a vital role in productivity and profitability', points out an important fact. Although the observation was directed at why search firms can play an important role in recruiting, it also identifies a new reality for MIS. Data processing has been placed in the spotlight with an emphasis on profit and loss.

In order for many MIS departments to have an affect on the bottom-line, they should have a specific business strategy to plan their activities against. Managers would then be able to assure a direct correlation between strategic planning and information needs. If the information needs were in turn driven by organization strategies, what helps us today, would continue to help us tomorrow.

According to William Darnton(Managing Partner, Anderson Consulting, Chicago, Illinois), '...you have to manage expectations as part of the systems appraisal process'. This seems to point out an important fact of life for MIS departments. In essence, be honest and realistic with your customers. This way, user areas won't expect a 'dream' system when they are only getting basic functionality. Also, the user will have a more realistic appraisal of task deliverables. Keep in mind that when a user says 'I need it now!', they may really expect to have it now!

User demands for service should inspire MIS managers to tune activities and squeeze all they can from their staff. MIS has a real opportunity to become a gateway for realization of corporate strategic visions. It can also lead to information sharing that would allow a company to leap into new territories, from a customer service perspective.

Summarizing the management role, it was perceived differently in the past and continues to evolve. It appears to present a challenging opportunity to flexible planners, if they are willing to play a visible role in success. Through careful observation, deliberate planning, tight resource management, and plenty of hard work, MIS managers can find a new definition of success in the 1990's.

IS THERE REALLY A FORMULA FOR SUCCESS?

Formulation of a success plan could easily be a consuming goal for even the most enthusiastic manager. Hopefully, one can readily identify some specific areas where adjustment could improve the opportunity to succeed.

Initially, a manager should attempt to clearly communicate the role of MIS as a participant in cost control and quality improvements. Care should be taken not to have the user areas interpret the role as an intrusion into user concerns. Once this is accomplished, a myriad of other points can be covered.

Playing a participative and supportive role in the strategy formulation process can foster user confidence. Special efforts should be made to protect confidentiality and to affirm faith in the MIS development process. MIS should strive to make every user an ally. Political entanglements can be minimized by inviting group contribution/consensus on particularly sensitive issues. Assistance should be given for defining where the user 'is' and what is important to keep doing well. The development issues that are key to strategy implementation can be identified and noted. Clear and concise communications regarding strategy analysis should be developed with fresh/open-minded thinking. Honest assessment of strategy implementation requirements becomes extremely important. The fact that it could take years to realize goals must be communicated early in the strategy assessment process.

After soliciting user-defined needs and a thorough review of corporate strategies, MIS can focus attention on their own role. Managers should begin by developing a solid plan for determining the types and quantity of services that users will require. The data/information that needs to be assembled and tracked for strategy for strategy support should be identified. Managers should also correlate the products, services, customer base, and market segments serviced into a logical systems development strategy. When all of these items are coordinated effectively, a meaningful and realistic user support/training strategy can be implemented. Every user should visibly receive something from the plan in order to maintain complete cooperation.

COMMUNICATIONS AND STATUS REPORTING

The MIS organization needs to establish reasonable and precise goals that will lead to strategy fulfillment. All performance should be periodically measured against goals in order to assess reasonability. Key benchmarks and baselines can be developed to clarify where MIS is, where it wants to be, and how it might get there. User areas can be kept aware of MIS strategies/progress based on pre-defined reporting intervals.

MIS organizations need to evaluate current and future software requirements, hardware needs, and human resource requirements. Plans for balancing haves versus needs should be developed.

Proven project management methodologies should be utilized or improved. An appropriate decision-making process that fits the methodology needs to be instituted. This process should also include uniform policies and procedures. The support structure implemented must be rigid enough to assure consistency, yet, flexible enough to permit free exploration of multiple alternatives for system solutions.

CONCLUSIONS

MIS resources should be evaluated by assessing its' responsiveness and correlated support of stated corporate strategies. Communications on development activities, especially progress, should be reported objectively and honestly. Only factual information should be included in progress reports. Positive points should always be presented first, thus keeping the emphasis at a 'can do' level.

Daring to measure yourself in an honest and visible fashion will help to convince users that you are working towards their goals/needs. Keeping an emphasis on positive progress reporting will also convey an important message--MIS can and will play an important role in corporate progress.

Is there really a formula for success in the 1990's? We can only hope that application of logic to our management roles' will enable us to make better decisions. If these decisions are focused on corporate strategy support, our chances of successfully making the techie to manager transition should be greatly enhanced.

1803
High Impact Presentations

James L. Bonner
MacMillan Bloedel Inc.
State Highway 10
Pine Hill, Alabama 36769
(205) 963-4391

HIGH IMPACT PRESENTATIONS

A quote I recently read made me think about this paper. It said, "Don't be afraid to go out on a limb - that is where the fruit is." When faced with a presentation, most of us need a little encouragement to go after the fruit. It is all too obvious that the limb gets smaller as we near the fruit. I would like to share with you some occasions when we managed to reach the fruit, as well as some times when the limb broke.

Several years ago it occurred to me that I might find some value from collecting copies of that special slide that was supposed to carry the day for the presentation. This has turned into a minor hobby, since every presentation builds up to some important point. I know from experience that the slides you will see have taken literally days to put together. Every word and every icon was chosen to convey just the right thought.

When you consider the importance we attach to our presentations, the time we invest in them, the cost of preparing the presentations, and how few chances most of us have to get them right, there is good reason for each of us to take another look at what went right and what went wrong.

Each of the following slides has been extracted from a real-life presentation within our company. Most of these slides were intended to be the "HIGH IMPACT" slide for that show. Some of them worked just the way we wanted them to. Others did not. I really think you can learn more from the ones that missed their mark, but my pride won't allow me to share just those with you.

In my opinion, our company's presentations have improved in format, if not in content. This has been directly related to our ability to learn to use PCs and their drawing software packages. We also have excellent low cost color printers and the option of overnight 35mm slides. We have come a long way from the typed page run through the copying machine on a transparency, or if you were in a real bind, the trusty grease pen.

Preparing this material can be fun for a creative person with a sense of proportion and color. The process is undertaken by all different levels within our company, from secretaries right on up through the mill manager. Sometimes it seems that we have raised the bar for colorful, consistent materials so high that the beautiful materials overwhelm the point - something like being overdressed for the occasion.

The real story, however, is in the slides themselves.

Stress & Burnout: An I.S. Connection?

Robert M. Gignac
Levi Strauss & Co. (Canada) Inc.
80 Allstate Parkway
Markham, Ontario
L3R 8X6

Could it be possible that your job makes you sick? Do you awake in the morning already exhausted before you start your day? Do you feel helpless about the way things are, and powerless to change them? Are you trapped and worn down by drudgery? If these feelings are familiar to you, or someone you work/live with, then you may be experiencing what some have called the 'modern epidemic': job burnout.

This paper will explore the nature of stress and job burnout, suggesting ways to avoid it (if possible), and offering some hope if you have started to succumb. We also will examine the link to the I.S. profession to see if we are at more risk than people in other professions.

Introduction

If a psychologist were to look inside the average DP shop, what would they see? In all probability, they would witness a high technology environment subject to rapid change, filled with achieving, independent thinkers, who are prone to low morale. In short, the DP shop is a fertile ground for psychological research.

Defining Stress and Burnout

The terms 'stress' and 'burnout' are often linked together, but they have entirely different meanings. It is believed that stress results from environmental situations that require behavioral adjustment - ranging from petty daily annoyances, to significant events such as job loss, divorce, or death of a family member. Originally introduced to the social sciences by Hans Selye in 1936, defined as 'non-specific response of the body to any demand'. We may more easily define it as 'any situation to which we cannot respond in an efficient manner'. Selye's description of stress focus heavily on 'adaptation', as our bodies attempt to counter stress. We can liken adaptation energy to a bag of coins. Once spent, they cannot be replaced. Each individual inherits a different quantity of adaptation energy, and uses it up at different rates. The common thread, is that once spent, a situation known as burnout occurs.

Burnout is the harder of the two elements to define as it is not necessarily a 'thing', but a state of being. A working definition might read, "a syndrome of emotional exhaustion and cynicism that frequently occurs among people who spend considerable time in close encounters." Burnout is a very unpleasant experience, usually preceded by a long period of high stress during which the individual has lacked control. Generally occurring in individuals who pour more into their work than they get back, once thought to be limited to those in the 'helping' professions (nursing, counsellors, doctors, etc.). This state is often characterized by chronic emotional and psychological fatigue, anger, self-criticism, cynicism, feelings of helplessness, apathy, depression, and disorientation. Even routine tasks such as taking a shower can become difficult.

An aside to this discussion stems from the results of personality test surveys on people having been defined as suffering from 'burnout'. The test results disclosed that these individuals had a higher need to do a job well for its own sake than did their peers, and they initially had a higher need for achievement rating. Reported studies on DP professionals repeatedly show that DP staff are prone to possess the same personality characteristics. We will return to this point later.

What causes Stress?

Stress can be caused by many things, including (but not limited to) the following: sensory overload/underload, marital disharmony, low self-esteem, feelings of inadequacy, lost status at work, unfulfilled achievement needs, role ambiguity, environmental influences, poor nutrition, poor time management, lack of a social support network, poor physical fitness, loneliness, unrealistic expectations of others, changing social norms, and communication overload, among many not listed here. In fact, stress can be a cause unto itself. As one researcher puts it, "Stress, besides being itself, and the result of itself, is also the cause of itself."

Stress is like ivy growing up a tree. At first, the tree hardly responds. The ivy grows up the trunk and digs into the weak spots in the bark. The ivy then starts to cover the branches and crowds the leaves of the tree.

Eventually, the ivy strangles the tree by cutting off its sources of light and moisture. Stress affects people in much the same manner. The problem lies with the cumulative factor, and our ability to underestimate the amount of stress we face. Any of the above situations alone may not pose much concern. We all at times have faced many of these and probably never thought about it. Yet, add several of those situations together, combine them with the wrong timing in terms of personal health, you could be on your way to serious problem. Of course, the corollary to this is that a little stress is in fact a good thing. The relationship between stress and performance can best be depicted by an inverted 'U', and is often called the Yerkes-Dodson law. First described in 1908 by Drs. Yerkes and Dodson of the Harvard Physiologic Laboratory, individuals perform at their best when there is a moderate level of arousal. If the arousal level (stress) is low, individuals are likely to be unfocused and unmotivated, conversely, the arousal level may be so high the individual becomes too anxious to perform. In this instance, stress can be the vehicle by which we mature, because it forces us to extend ourselves, and it is this extension that allows us to expand our abilities. The goal of stress management would then be to limit the harmful effects of stress and enhance the positive aspects.

How can we spot someone under stress?

In any individual suffering from severe stress there are certain telling signs. Unlike physical illness, the signs may not be obvious (ie: no outward expression of pain). The following is a list of possible signs you may encounter in individuals under stress:

- Declining productivity, both in the quality and quantity of their work
- Working later, more obsessively than usual, perhaps harder than seem appropriate for the situation
- Absenteeism or chronic lateness
- Making the safe choices, not the best ones
- Constant repetition of the same subjects at meetings, especially noticeable if the point is not particularly important
- Inappropriate anger, hostility or outbursts of temper
- An efficient worker suddenly becoming careless
- Tendency for team players to suddenly work alone
- Excessive worrying, especially over trivial problems (or, extreme casualness in the face of a real problem)

- Confusing or forgetting dates, places, meetings, etc.

In short, any noticeable change from the usual patterns of behaviour or performance, can be an indicator of someone under stress.

The Burnout Scenario

Like the human life cycle (or perhaps more appropriate, the system's life cycle), burnout goes through distinct stages. It begins when your stress responses mobilize your body so frequently in the face of chronic stress, that you deplete your energy available for adaptation. This situation can happen after six months on a new job, or fifteen years on an old one.

The process has been separated into five distinct stages, each being more serious than the preceding one. However, I should point out a few things. First, the theory I am outlining is not the only theory surrounding the concept of burnout. Second, no two people experience burnout in the same fashion. One might remain in the early stages for many years, then progress rapidly to the later stages. Others might move quickly through all five stages, recover, then remain free from burnout symptoms for the rest of their lives. As well, wide individual variations exist in the symptoms of burnout. It is these variations in progression and symptoms that make it easy to ignore burnout and treat only the visible symptoms, rather than the actual cause. Burnout is a complex and elusive condition. No laboratory tests can reveal it. To learn about coping with burnout, we should examine the stages and the symptoms we might expect to find at each.

Stage One: The Honeymoon

The vast majority of workers start their working lives with both enthusiasm and desire to work, enjoying the exhilarating stress of learning, meeting new people, and showing their competence. The enthusiasm comes in part from their desire to succeed. Nobody sets out to be a failure, especially after only a few weeks on the job. As the honeymoon stage continues, we often feel perfectly matched to our jobs. Any stress that we feel only gives meaning to our work and adds to the sense of fulfilment.

Two important things take place here. First, despite our enthusiasm, even enjoyable stress uses up adaptation energy. Second, it is in this stage that we develop our habit for coping with stress. If we are successful, the honeymoon stage may proceed without end, and for many people it does. Should the strategies we develop be ineffective, the burnout process beings in earnest. Perhaps more importantly, we miss the opportunity to equip ourselves to deal with the stress that faces us not only from our jobs, but from life in general.

Stage Two: Fuel Shortage

To many people, fuel shortage begins with a vague sense of loss. The honeymoon is definitely over, generally with a feeling of contempt toward your job. The challenges of your work and your enthusiasm for it has waned. People caught in stage two may be called 'tightrope-walkers' because they are continually on the verge of losing their balance. Just as unrelieved stress build up to push them off the wire, they immediately regain their balance.

At some point late in this stage, many of the most frequent symptoms of job burnout make their appearance: job dissatisfaction, inefficiency at work, fatigue and sleep disturbances.

Stage Three: Chronic Symptoms

The vague sense of loss encountered at stage two has now turned into 'something is happening to me.' Physical and psychological symptoms are becoming more pronounced, and it will be harder to dismiss them. Whatever symptoms affect an individual, while varying widely, will be more difficult to shake off. Most common to this stage are: chronic exhaustion, physical illness, anger and depression.

As the third stage progresses, people often find themselves with a perpetual feeling of anger. Almost anything can cause them to erupt, and people begin referring to them as someone with a 'short fuse'. The change in most prevalent in someone who, in the past, has been easygoing, calm and accepting of ideas or criticisms. As burnout progresses, the general anger often becomes more focused, and centres on one or two individuals who are close to them.

Others avoid the problems associated with anger only to end up suffering from depression. This depression can have numerous causes, and stress is only one of them. Unfortunately, depression is a common response when some aspect of work becomes intolerable. They can't quit, they can't fight it, and it just won't go away. The harder people try to remove themselves from the situation, the worse it gets.

Stage Four: Crisis

For the vast majority of us, we will never progress to stage three type behaviour, and we should count this as a blessing. As burnout progresses into stage four, symptoms get worse, and recovery time lengthens. Four significant changes occur at this stage: physical symptoms become critical, obsession with frustration begins, pessimism permeates thinking, and an escape mentality develops.

Obsession turns people into moody, withdrawn individuals. Specific topics dominate all conversations with this individual. Their jobs, and sources of stress refuse to leave their thoughts, they talk of nothing else, and they may even have nightmares about the problems. Those around them feel powerless to help, and for the most part, they are. In time, the obsession with problems will lead to self-doubt and pessimism. Competent decision makers will hedge their bets waiting for more information. Second guessing becomes second nature, and the poor decisions only add fuel to the fire.

A clear progression can be seen in the first four stages of burnout. Satisfaction with work and exhilaration marks the honeymoon stage. Initial stirring of dissatisfaction begins during the fuel shortage stage. If our safety valves don't work, symptoms may begin to be chronic. Frustrations and discontent usually focus on aspects of their job or career. In the crisis stage, self-doubt and despair blanket perception. Often wishing for a convenient escape hatch, the individual can feel like a trapped animal.

Stage Five: Hitting the Wall

People who are familiar with distance running, especially marathons, will be able to identify with stage five. It is said that a marathon really starts at the twenty mile mark, the last six are the hardest. 'Hitting the wall' is an experience so devastating it can take someone completely out of the race, paralysing the body to the degree that all physical control is lost. In the contest with stress, people who have hit the wall find themselves unable to work, and perhaps may not be able to for years. For the unfortunate few who reach this stage, burnout has become so entwined with other problems such as alcohol, drugs, and psychological disorders, it cannot be easily untangled. Recovery from this stage will elude some, others may win the battle, but it will require time and extreme patience from those close to them.

It is not inevitable that we progress through all the stages of burnout. We must accept that stress is a situation that must be dealt with, and by learning to cope, preventing the process, we can recover should any damage be inflicted. Knowing the stages in the model is only the first step in dealing with burnout.

Is there anything we can do?

Yes there is, and it's called: Stress Management. Stress Management is the process of taking those elements in your life that create stress for you and learning to: 1) adapt to them, 2) reduce the severity of the stress these elements cause, or 3) learn to avoid these stressors if the first two cannot be accomplished. Stress Management will not eliminate stress from your life. First, it would be next to impossible to perform such a task. As desirable as it might seem, removing such stressors as traffic jams, pollution, office politics, work deadlines and overcrowding, it is just not possible. Second, even if we could accomplish it, eliminating stress is not desirable as some stress is a requirement for daily living. Stress provides us with the maturation to accomplish tasks and to change things.Stress can extend our abilities to the point that we often do our best work under non-debilitating pressure. Many examples of this are seen in the sporting world, where records are rarely set in routine practise sessions, but often under intense competition (ie: the Olympics or World Championships)

What steps should we take to help us manage stress? We will focus on two: Content and Concept.

Stress Management - The Content

The content of stress management is the quantifiable portion, filled with specific tangible data. On a lighter note, it is also the stuff our mothers have been telling us since we were teenagers. Exercise, rest/relaxation, and developing good appraisal skills are the keys. Not only is the content good for us, it works. If we were all to exercise regularly, eat a healthy diet, take proper vacations at appropriate intervals, besides having regular medical checkups, we would all be much better off. Well, if it's so easy, why don't we do it?

Time, lack of motivation, fear of the unknown perhaps? The reasons we choose to avoid doing what is intrinsically good for us are as varied as our individual personalities. We must breakdown the barriers that prevent us from helping ourselves.

Exercise

The mere mention of this word is enough to make many people break in to a sweat - a cold one at that. Exercise conjures up images of hot steamy weight rooms, grey flannels, and pain. I'm here to tell you that it isn't so. Exercise doesn't have to be life's sole obsession to be beneficial. The Nike shoe company has been doing us a big favour with their ongoing ad campaign "Just do it!". Walk, jog, swim, bike, play tennis, skip, any sporting activity that brings you pleasure.

If I hear any complaint about exercise, it has to be the major complaint of life today - "I don't have enough time!". When it comes to time, all things are relative. If you'd like to have an Arnold Schwartzenegger physique, 15 minutes a day won't do it. For the purposes of making us a little less sedentary, and increasing our energy levels, 15 minutes a day would be a great start. "But I can't get away from my desk", well that's ok, you can start while you're sitting at it. Keep a pair on sponge rubber balls or exercise handgrips in your desk drawer. On hold while on the phone? Use them. Taking a five minute break while think through some design problem? Use them. Another useful office technique is to keep a pair of 'softweights' in your desk. Wear the ankle weights as you walk around the office. The walking you are doing is beneficial as it is, the extra resistance caused by the weights will make the movement even more beneficial

In short - "Just do it!". Remember, every little bit helps.

Relaxation

A popular piece of advice runs, "Just relax. Keep cool. Take it easy". In reality, this is often easier said than done. Many people complain that they don't know how to relax. Often the word 'workaholic' describes them well - people who are so preoccupied with their work that even their vacations are stressful.

Relaxation is a skill and as with any skill it requires practice. Audio tapes are available that will allow you to practise such techniques as relaxed breathing and muscular contraction. As your skills develop, relaxations can be attained through deep breathing, and this skill can be used to successfully battle stress on the job. Mental relaxation is also a key part of the process. Some relax by engaging in strenuous exercise (see preceding section), others by becoming absorbed in favourite hobbies or tasks.

Learning to appraise situations

Appraisal or perception is an important element in determining how we view stressful situations. 'Cognitive Appraisal' refers to an individual's judgement of the demands and constraints in any given situation and the options at their disposal for dealing with them. Appraisal can be viewed as a two stage process - primary and secondary.

In primary appraisal, we view the situation to be either relevant or irrelevant, challenging or threatening, os possible benefit or harm. These decisions are often made without our thinking about it - at an almost unconscious level. Primary appraisal answers the question, "Am I ok, or am I in trouble?". How did the person on the other side of the desk approach with this problem? Was it, "Hey Bob, do you have a minute?", or was it, "BOB, WE HAVE A SERIOUS PROBLEM, THE MACHINES ON FIRE!!". Our view of an event, as threat, potential loss, or challenge, will determine the degree of our stress reaction. Personal beliefs, goals, values and commitments will all factor into our primary appraisal of events.

In the secondary appraisal, we make judgements about the adequacy of our available coping skills, as they pertain to the situation. Here we answer the question, "What do I do about this situation?". We evaluate our coping strategies in terms of their cost, and probability of success, based on factors such as past performance, self-confidence and material resources available. Like primary appraisal, this too, occurs automatically. As our attempts to cope with a situation cause the situation to change, we repeat the process in an endless cycle until the situation is resolved.

Areas for us to work on in the appraisal process centre around our ability to see situations for what they really are. Perception is in fact reality. When we view a situation, we act upon it based on its appearance to us, even if we see it in the incorrect light. Taking the time to realistically appraise situations can ensure that we make better decisions when we act upon this information. If possible, try to view the situation as an outsider would, or by perhaps altering the viewpoint of the problem. Development of empathy with people you deal with regularly can help to ensure you view situations in the best possible light, ensuring a higher degree of appropriate reaction.

Stress Management - The Concept

The concept of stress management is the second half of a cohesive model to deal with stress. The content discussed earlier are things that we need to do to make us feel better - but if we abhor exercise, despise fruits and vaguest (even President Bush won't eat broccoli) all the extolling in the world on my part won't make you change. Keep in mind that content alone won't solve your stress problems either. In this section we will focus on the personal side of stress management: power, difficult people, energy, releasing problems, and maintaining balance.

The concept of power

Some seem to have it, some don't. Power is a main criteria in dealing with stress, because those who have it are generally under less stress than those who have none. Powerlessness in a situation increases our stress because we feel that actions and consequences are our of our control. Power is not a function of position, despite what are organization charts suggest. People obtain/lose power as a function of their relationships with others. This is why children often have power over parents. Dependency gives people power because once someone is dependant, the person feeding the dependency must continue to do so, or incur the wrath of those dependant on them. Powerlessness is regularly described as having a lack of available choices. Ask someone who puts up with a poor work situation or bad relationship and they respond with - "I don't/didn't have any choice". The truth, however ugly, is that we always have a choice, often many, but we invariably don't like them. When we don't like the choices available, we don't seem to have any choice. To regain power in stressful situations we must make the difficult choices, even the distasteful ones, because they are often the key to removing ourselves from a stressful situation.

Difficult People

In variably, our workplace forces us into personal contact with others on a daily basis. Most of this contact will be interesting, enlightening and make work enjoyable. There will be other people that can make our work unpleasant, or even unbearable. Let's call them 'difficult people'. Rather than setting our phasers on 'stun' and cluing them in, how can we deal with difficult people? First, we have to realize that they often cause us to be powerless in stressful situations. Not because they took our power away, but because we give it to them. Second, realize that you can't enlighten unconscious people. We see them as irrational, egocentric, controlling types that 'just don't get it'. Are they conscious? Of course. Do we see it? No. To make matters worse, the more burdened we become, the worse these people appear to us. Under stress our perceptions change, and remember that our perceptions are indeed the reality that we act in.

Take as an example the habitually late employee. Management workshops teach us to document this lateness and we invariably confront them with the same. Faced with it, they will generally choose to ignore it, or give you 'solid' reasons for each occurrence. They may try to turn it against you, "Don't you have anything better to do with you time?", or even better, "You were late twice this month, what's your problem?". Key to the issue is because people can't solve problems they don't perceive they have, you can't help them by taking on the problem for them. As we will see later, ownership of a problem is the key to starting to solve it.

Energy

Spend some time focusing on how we gain, lose and use energy on a daily basis. Envision your energy total as a pie, you are allowed to cut it any way you like, but you only get one pie per day. After the details of work, commuting, home, play, you'll find that you rarely have much, if any, left for yourself. The key therefore is to spend you energy wisely. Spend your energy on conscious people, on concrete tasks and goals, and always reserve some for yourself. Most of us spend a great deal of time and energy on people who create problems for us. By focusing on the problems at work and on difficult people, we end up with less energy to give to those who are important to us.

Values

Values are a personal barometer that we use to define our lives. There are many, but for the purposes of this discussion, we will use three: Work, Home and Health. Draw a chart containing these values on a sliding scale as seen below:

WORK	HOME	HEALTH
MONEY	FAMILY	SANITY

Examine the pairs of objects and assign each pair a number from 1-3, with #1 the pair you value the most, #3 the pair you value the least. If you follow the majority of the population (but don't be dismayed if you don't), your order would be Home, Health and Work. Circle the pair you value the most. Now, put a box around the pair that gets the most of your time/energy. Where did that box go? For most of us, probably work. To support work as a value, we must violate the things we value the most.

Our primary values are the ones that we support behavioral. By this, I mean with time and energy. Situations where our primary values are not the ones we are supporting behavioral is known as a 'values bind'. To support a primary behavioral value, we will temporarily violate some or all of our secondary values. To support work, we violate home and health, we work late, we work weekends, we do home tired. To support home, we must spend less time and energy at work.

The best way to keep values in perspective is to rotate them. Make work a primary value, then shift it to home, and you will find that you are violating work to support home. Easy in principle, very hard in practise. Employer's don't want us to violate work to support home. Other times, it is our own need for approval that stops us from rotating values. Many of us have a difficult time saying 'no' to people, or dealing with the disapproval we encounter from those we have said 'no' to. Our need for approval can force us to make bad decisions. Remember that if you try to make everybody happy, you'll end up making nobody happy - including yourself. We need to make decisions that support our values - cut your energy pie in such a way that we do this. Problem? Of course. We invariably let other people cut our pie for us. How can we stop this from happening?

Dropping problems that aren't yours

Earlier I mentioned that we can't get people to solve problems that they don't have. Conversely, we often take control of problems that have nothing to do with us, thereby creating additional stress for ourselves. Generally, it is due to our refusal to let go of problems once we have them. How do we end up with them in the first place? In much the same way that researchers catch monkeys in the wild. Imagine a box consisting of narrow bars that would barely allow a monkeys's hand to get through, the turn the box upside down over a pile of bananas. The monkeys run in and grab the bananas, but the box won't allow them to get the bananas out. The researchers then go over and pick up the monkeys. Why don't they run? Because once they grab the banana, they won't let go. If the dropped the banana they would be free to run, but they persist in hanging onto it, and are thusly caught. We get caught hanging onto problems in a similar fashion.

When we attach ourselves to problems there are two issues to consider: 1) do we identify with the values entwined with the problem? If 'yes', then take it on. 2) Is this problem in an area of 'unconsciousness' (reflex, instinct or guilt) for us? If 'yes' then drop this problem. If that means we have to drop our attempts to change a co-worker, subordinate or spouse, then do it. The odds are always against you anyway. In business, as in your personal life, you can't get people to solve problems they don't have.

If you are tired of doing a co-workers job, or a tasks that never should have been yours in the first place, then do so. Be prepared to see some consequences, especially the negative ones. People are never happy when tasks go undone, especially the people to whom the fingers start pointing when it becomes obvious it should have been their job in the first place. If pressed for a reason, "I'm not doing it because I need not to", is sufficient, if in fact the task never should have been yours to begin with. However, if you make a conscious decision to continue doing the task and do nothing about it, then you must do the rest of us a favour, and stop complaining about it.

Maintaining Balance

Take a comfortable position, feet shoulder width apart, knees slightly bent. Image that stress is about to push us over. The more balanced we are the better prepared we are to cope with stress. The more balanced we are, the larger the amount of stress we are able to face before we begin to move. When we are standing on one leg, it takes little, if anything, to push us over. Difficult people (the ones I mentioned earlier), are often the ones able to get us off balance regularly. These same difficult people are often in good balance themselves, and once we got off balance, they have the upper hand. By doing this, they get our power. Our imbalance problems are often compounded as we often become 'verbal' as we start to lose balance. The discussion that become an argument, often including the phrase "I'm the boss here...". Don't worry, you are the boss, they know it, and you know it. The fact you had to tell them, should tell you something. When we are off balance and verbal, we will often end up saying things we will only regret later.

Stress Management is learning how to live with the situations when we are off balance until we can regain our feet. Our workplace seldom changes - even if our employers did us the favour of firing all the 'jerks' we are forced to work with, it wouldn't help. They would only hire more to replace them. Perhaps the best comment I have ever heard about dealing with difficult people came from my Grandfather, "The worst part about working is that you spend your days dealing with little boys dressed up as men". You do have a choice: learn to keep yourself in balance, or be prepared to live with the decisions you made while you were not.

A Dozen Quick Tips

When I first started to put this paper together, a friend remarked that it probably wouldn't make any difference what I told people about stress management unless I could tell them that it was effortless and would take effect immediately. He felt we have fallen prey to the 'microwave mentality', we want it hot, we want if fast, we want it now. As I have no 'microwave' solutions, I hope this section will suffice.

As those of us with an interest in social psychology are prone to do, I tend to spend some time perusing bookstores and libraries checking out the latest and greatest in the field of stress/stress management in the workplace. The variety of available information is endless. While I can't say that I've read every available piece, I have read enough that I have compiled a 'best of' list of one dozen good tips that should help you deal with stress on the job. While not every one will apply to each individual in all situations, I feel they offer some valuable suggestions.

1) Realistically appraise your current/future abilities. People sometimes overestimate their abilities and expect too much from themselves and those around them. Always strive for your highest obtainable goal, but learn not to resist it in vain.

2) Learn to laugh. Research has shown that when people laugh, it's often therapeutic. Have the ability to laugh at yourself. Consider the following from Gerald Weinburg, author of "The Psychology of Computer Programming":

"Last among the essential personality traits for programming, we might list a sense of humour. The computer doth make fools of us all, so that any fool without the ability to share a laugh on himself will not be able to tolerate programming for long. It has been said that the programmer's national anthem is "Ahhhhhhhhhhhhhh". When we finally see the light, we see once again that we have fallen into some witless blunder, some oafish practise. Only by signing the second verse "Ha ha ha ha...", can we long endure the role of the clown..."

3) Accept that there are some things that are out of your control - traffic jams, a failed hard drive, electrical outages during year end processing - some things are unlikely to change despite anything you do.

4) Don't procrastinate by doing low priority tasks because they are fun and easy - thereby neglecting high priority items until they become major crises. In short, practice good time management.

5) Work at building supportive relationships both on and off the job. This takes time, but those with supportive relationships to fall back on often suffer fewer consequences under stress than those who are socially isolated.

6) Learn about the sources of stress (some of which you've done by reading this paper). It is hard to accept that stress sometimes comes from our own inability to handle life's disappointments.

7) Stressful events are in the eye of the beholder and subject to the possibility of misinterpretation. Check your assumptions and perceptions to insure they are correct, and try to see the whole picture before reacting.

8) Avoid chronic hurriedness. High stress personalities often try to do too many things too fast. Tell yourself that no enterprise ever failed because it was executed with good judgement and attention to detail.

9) Learn to eat nutritious food. Unfortunately, it is easy to develop a habit of eating those things that satisfy our palate rather than those that satisfy the body. Even if this doesn't immediately reduce the stress levels you are under, your body will

thank you for it.

10) When stuck in traffic or waiting in endless lineups, use the time to be by yourself rather than fuming about the frustration. Visualize a more peaceful, tranquil situation.

11) Learn to feel comfortable talking about your problems. A close confidant is an invaluable resource during stressful periods. However, you must avoid dumping on the same individual all the time. Another ground rule us that you must be available for those same people in their time of need.

12) Allow for plenty of rest for mind and body. There is just no substitute for rest and relaxation. In addition, those around you may find that the rest does wonder for your disposition, thus reducing their stress levels.

Summary

This paper was created to present some basic outlines regarding stress management. As such, I may have generated more questions for the reader than answers. To help address these issues I have provided an additional reading list for those desiring more information. I hope that this paper piqued your curiosity about stress management, and that you can now commence to create a reduced stress environment to work in. Stress management may be seen in some respects as self management, and we owe it to ourselves (and those we care about) to take control of stress before it takes control of us.

PAPER NO.: 2001

TITLE: The Communication Puzzle - Making It All Fit

AUTHOR: J.C. Strote
KVIE Channel 6
11000 E. Jahant Road
Acampo, CA 95220
916-923-7474

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO.: 2002

TITLE: Cost Justification Basics

AUTHOR: Robert Andrews
Dynamic Information Systems Corp.
652 Bair Island Road #101
Redwood City, CA 94063
415-367-9696

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

HOW TO TELECOMMUTE AND RETAIN YOUR SANITY

PAUL EDWARDS
PAUL EDWARDS & ASSOCIATES
1506 ESTATES WAY
CARROLLTON TX 75006
(214) 242-6660

HOW TO TELECOMMUTE AND RETAIN YOUR SANITY

ABSTRACT

Telecommuting is becoming a very popular form of employment. There are many factors to be considered to make this new working environment successful. If the telecommuter is properly prepared, then this can be a very efficient situation for the employee and employer. The telecommuter could work part-time or full-time. Extensive requirements definition are a must.

Proper equipment, software, office environment, work attitude, and comprehensive employer policies and procedures are vital for success.

An in depth discussion of these areas will be provided and practical examples of personal experiences will be presented.

Telecommuting is a fast growing form of employment. It can be used effectively by full-time employees of companies or by part-time entrepreneurs. We will discuss the particulars of this work environment in depth. The amount of success enjoyed by the employee is directly related to good planning with realistic goals in mind.

This presentation covers the definition of telecommuting, justification, equipment, software, office environment, work attitude, and employer policies. Most of the material presented is based on many years of experience in this field.

DEFINITION: The word telecommuter is made up of two words, TELE and COMMUTER. Webster defines TELE as far off, distant, remote/at, over, from, or to a distance. And COMMUTER as a person who travels regularly between two points at some distance.

The corporate employee may be telecommuting on a part-time or full-time basis. The determination of type is based on the justification needed.

JUSTIFICATION: Sometimes the talent required for specific tasks is not available locally. This talent could be made available to a company by a telecommuter. A personal situation of an employee, such as pregnancy leave, illness in the family, or handicaps or other special cases of valuable employees can provide the justification needed. The transportation costs or restrictions of certain areas of the country may make telecommuting necessary.

EQUIPMENT: Office machines, computer systems, and furniture must be purchased or provided to properly set up the office environment.

Basic office machine inventory should consist of a telephone, fax machine, and copier. The neighborhood printer or office supply store might provide the fax and copier services. This should only be a temporary solution. You will waste too much time, otherwise. A two line phone switch can be installed to answer the home and business lines from one instrument. A headset is very valuable if much time is spent on the telephone. The basic fax machine should have an automatic paper cutter, 10-page automatic feed, speed dialing, and Group 3 protocol.

The portable or cellular phone adds flexibility and movement to leave the home office without missing calls.
Voice mail or an answering machine can assist with the answering of the telephone and taking messages to prevent an important message from being missed.

A minimum configuration desktop PC should consist of a 2mb main memory, 386SX based CPU with 40mb hard disk storage and 1.44mb floppy disk. VGA monitor prices are reasonable. A laptop PC is a necessity for traveling use. Compatibility with the desktop PC is a requirement. For any remote access, a 2400 baud modem on either PC is a minimum. The price on 9600 baud v.32 modems is falling rapidly.

A minicomputer is very affordable if it is needed. A small HP3000/37 can be purchased for about the price of a PC. If you do your own support, it is a very inexpensive system.

The printer requirements will determine the type and number of printers. They range from inexpensive portable inkjet to dot matrix to laser page printers. A laser printer is the preferred choice for quality, speed, and low noise level.

Furniture can be obtained from family sources, used office furniture stores, computer stores, and large furniture outlets. The basic office should contain a desk with a good back supporting chair. The standard desk may be too high for comfortable access to a PC keyboard. A computer table should be used instead. A thorough analysis of work flow will determine the placement of equipment. If other individuals visit the office, sufficient seating should be available for their use.

A sufficient supply of office materials can be supplied from the supply room of the company. You will need pens, pencils, paper, disks, and any other items used in the office.

A complete set of manuals are needed for any hardware and software system used. Some software packages offer the option of printing manuals from the programs themselves.

Flexible storage must be abundant and easily available.

Shelves can be wall mounted or located in closets. A four drawer file cabinet is needed, also.

SOFTWARE: Basic software considerations are operating systems, word processing, spreadsheet, graphics, data base, communications, language compilers, and electronic mail.

Ensure that any software purchased is widely used and supported by all equipment you have.

The usual operating system for the PC is MSDOS but UNIX is gaining in popularity. If you purchase an HP3000, current FOS and subsystems tapes should be provided. Windows for the PC provides a graphic interface to isolate the user from the operating system. A mouse should be used with Windows to derive the most benefit from the interface.

Word processing software needs are driven by the business focus. The needs will vary from simple letters to programming language text to document processing. The popular products are WordPerfect, MS Word, Wordstar, and PageMaker.

Many people use the Lotus 123 spreadsheet product. There are several compatible products available. Compatibility with other software and hardware is important.

Graphics capability are included in some of the spreadsheet and word processing software. Harvard Graphics provided a very simple way to make the slides for this presentation. It will also make a slide show with your slides for a moving presentation if you use an overhead projector display or a PC projector. High quality graphics are essential to a professional presentation.

There are many data base products available. The best known is DBase IV. Most are relational with menu-driven interfaces. If a fourth generation language is used with the HP 3000, such as Speedware/4GL or Powerhouse, there is the capability of using them with a PC as well as the HP 3000.

Communications products must be chosen based on the systems to be accessed. If access to an HP3000 is required, then Reflection and Business Session are the most popular.

Requirements for programming will determine the language compilers needed. Microsoft has Cobol, Basic, C. Micro Focus has a main frame Cobol. Borland has Pascal and C.

Electronic mail systems vary from simple bulletin boards to extensive products like HP DESK/HP MAIL.

OFFICE ENVIRONMENT: The room used as the office is very important to the telecommuter. Location, power, air conditioning, lighting, and communications are major parts of the successful installation.

A room shared with other home activities will function temporarily but should not be considered permanent. The ideal is a spare room, such as a bedroom or basement is preferred. To limit noise and other outside interference, a room with doors to shut out the rest of the house activity should be used. A room could be added on the house if the funds are available and the existing structure will support it.

Power for all system devices should be wired directly to dedicated outlets with 20 amp breakers. Air conditioning, if required, should be provided to maintain a proper temperature level year round. Lighting should be sufficient to ease eye strain. Communications lines should be available for voice and data communications. Voice/fax/modem switches are available to share lines to cut cost.

WORK ATTITUDE: A successful telecommuter must act as if he is at a formal office. Mood is very important. You need to set work hours that are realistic. You must be a self starter because no one is there to manage your time and work habits. Also, you must resist the temptation to be a workaholic.

You should dress based on the interaction with others and at least have some basic formality, not a bathrobe. You will tend to wear out tennis shoes and blue jeans instead of suits and dress shirts.

You will find a need for social interaction. You don't have the office water fountain to congregate around. This could cause large phone bills to satisfy the urge to socialize.

The support that is given by the family is most important to your success. They have to agree to the fact that you are at the office for specific times during the day even though you have some flexibility. Stay involved with your local users group.

EMPLOYER POLICIES: The employer can save costs by having you telecommute. You need to have a written agreement for expense reimbursements, attendance at meetings, time and progress reporting, capital equipment purchases, and insurance. This agreement provides the formal ties to the employer. It also delineates responsibilities. This is important to the corporate employee who works part-time or full-time.

A corporate employee should have a business plan for their telecommuting operation. This helps set priorities and provides economic justification, if needed.

2004

SalesSpeak - What They Are Really Saying

By

Jeff Franz

Of

21st Century Systems Group

12475 Woodford Way
St. Louis MO 63044-1368
(314) 298-9659

Language is a software sales person's number one tool. Correct use of words and phrases can mean the difference between success and starvation. For the software buyer, understanding this language can mean the difference between getting what you want and getting what you were sold.

Most sales people will not deliberately mislead a prospect. They know that repeat business is important to their livelihood. Also, since sales depend upon references, it is in the sales person's best interest to be honest.

It is also in the sales person's best interest to make his or her product look as good as possible. It is their job. This brings us back to language. They will use every tool possible to do their job.

The words and phrases commonly used in the software trade may be called Salesspeak. Listed below are some examples of this language. Also included are the real definitions and some possible corrective action.

SalesSpeak - The Language

It is Possible - This phrase is almost as over worked as the word "very." What this means is that under the right set of circumstances, the stated feature will work. What those circumstances are and what else needs to be done is a great unknown. More information is needed and should be obtained. Remember that it is possible to jump from an airplane without a parachute and live. It is possible, but not workable on a routine basis.

We can do it - Of all the phrases associated with SalesSpeak, this is the most common. The operative word is *can*. It does not mean that it does it *now*. Its severity can range from zone 1 to zone 10. Zone 1 states, "the system handles this without modification." Zone 10 is the statement, "if I can convince the president of the company to put a user exit in this place, you could write a subsystem to do it."

In order to determine this statement's position on the continuum, watch the sales person. If you detect nervousness, you are nearing zone 10. You should approach this zone with caution.

Even zone 1 can be a problem. You must determine that the system handles this *in the way you need it*. Just because it does it, does not mean it does it your way. A car can function as a can crusher but not very well.

When this phrase appears, the word "how" should immediately follow from your lips. If it takes response conditioning on your part, it is worth it. Once an explanation is given, become a Missourian, say, "Show me." If this is an important feature, do not buy the software until you see it to your satisfaction.

Let me check on that - This can be a very deceptive phrase if used correctly. It appears that the sales person simply does not know the answer. The truth may be that he knows the answer and it is bad. Why does he not say it? There are two reasons.

The first reason is that the sales person hopes that someone at the office will dream up a way to accomplish the same thing. This can be called the Anne Sullivan ("The Miracle Worker") approach. This is a completely legitimate reason as long as the sales person gets back to you with the correct answer.

The other reason is that he hopes you will forget about this question. It is remarkable that some potential customers will not note this for follow up. The solution to this problem is to note all unresolved issues for later response.

Flexible - Flexible is an all time favorite word among software brochures and sales people. It seems that every thing is flexible. This sounds too good to be true. It is.

This word means that a wide range of interests must be served. It also connotates compromise. The Swiss Army knife is a flexible tool. It has a scissors, knife, can opener and other necessary devices. However, it does not make as good a scissors or can opener as the real things.

The only way to determine whether a "flexible" piece of software will work for your particular application is to contact references. These people must be in the same industry with similar needs. If the software vendor cannot supply the right type of references, begin amending the license agreement. Include necessary functions and features as binding conditions.

User - What is a "user?" To the customer, a user is someone outside of data processing that works with an application. The software vendor considers a user anyone that uses the application. This includes data processing people.

The sales person likes both definitions and will use the one that best suits the situation. Therefore phrases such as "user controlled" and "___ by the user" should be viewed with a careful eye. Even if the sales person is referring to the customer's view of user, it may not be desirable for the user to control it. You should ask many questions.

User Friendly - This is a special subset of the above. Besides the differing definitions of the word "user," all users can be lumped into one category. Therefore, a user friendly piece of software might be easy for an advanced accountant but unusable by a clerk. It also may be user friendly for a programmer with experience in machine level languages.

Normally a demonstration can expose the pretenders. It also may take some digging to be sure. You should be ready to spend some time looking at the functions and having the appropriate people try them.

Available - They might as well say, "extra cost." The word by itself means the same thing. If it is not extra cost, you will by all means know it. As long as you are aware of the meaning, this is normally a harmless word. The only possible follow up questions are, "From whom," and "How much."

Optional (method or feature) - If the sales person wants to attempt to hide hidden costs, he will use the word "optional." Automobile dealers use the word to denote extra cost. Software dealers can use the word to mean extra cost or free. Every time the sales person says this word, ask, "how much?"

It is also wise to find out more details about this optional feature or method. This feature or method may be more a more cumbersome alternative.

Alternative method - If you have received an answer from the sales person and this phrase follows, watch out! It can mean one of three things.

The first meaning is that the sales person has found a better way to do something. He should be questioned on the benefits of his better way. Aspects such as system performance, manual effort and reporting ease should be considered. This is a legitimate answer.

The second meaning is that the system is not efficient as it was first described and the alternative method works easier. It is different from the first case since the initial method is available but it does not work very well. This method should be explored in the same manner as the first.

The third definition is trouble. The system does not handle the desired way and the sales person is hoping to high heaven that you choose the alternative. If you are trying to catch the sales person in one of these, have him show you both methods. Under the first two meanings, you will see something. If the last meaning is true, you will know it. You should notify the vendor of your dissatisfaction with the sales person.

The next release - When is soon? This is the question to ask yourself when the statement "in the next release" is made. A worse case is the next version. These statements normally appear when you want to see something that is not in the software. It means, "I heard someone at the office talk about putting it in so it must be going in."

Both of these timing elements provide challenges if you cannot see what you want. The next release is normally in process and can have some date associated with it. Depending upon the number of changes to be made in this release, the date may be relatively firm. Ask to see a list of what will be in the next release and a status of each. Be sure your enhancement is on it and that it has a status other than "under consideration" or "under analysis."

If your enhancement is in the next version, be careful. This can be a double edged sword with poison on the handle.

The first problem is timing the next version. You will want it as soon as possible. The software company may have other designs on the programming time it takes to do it. If you force the software vendor via the contract, they may rush the enhancement to you with little or no testing. This can be a disaster. On the other hand, not forcing the vendor may result in the feature being dropped. Features "in the next version" should not be important.

In Beta - This is a more definite time frame than the previous one. It also can have its problems. The "Beta" sites might actually be "Alpha" sites. Therefore your function is longer coming than you thought. Obtain the names of the Beta sites and contact them. Ask them how the testing is going and how the vendor's response has been to problems. This will give you an idea of the function's readiness.

The software vendor may not want to release the name of the Beta site because of past problems. Ensure the vendor that you are trying to determine the date of general release and will not consider past resolved problems.

Full featured - It is hard to find a more ambiguous term than this. (Unless you consider "performance," below.) What is full featured? The vendor will want you to think it has everything you could imagine. The package may offer many features. The features may be useful or they may be "bells and whistles." What ever they are, they can be dangerous.

How can full featured software be dangerous? Consider that all software is a series of paths. The software controls these paths via statements (conditions). The more features that a given piece of software has and the more flexible it is, the more paths it contains. As the number of paths increases, the chances of testing all these paths decreases. The decrease in the percentage of testing increases the chance of bugs. Therefore, more features can mean more bugs.

Another item to consider is that one customer's feature may be another customer's bug. If you have purchased software to work one way, it may disappear in a future release. Many vendors will install "switches" to allow a given function to work two ways. Be sure that this is the vendor's policy.

Easily Interfaced - A square peg can be easily interfaced with a round hole given sufficient quantities of sandpaper. There are two things to remember about interfacing. One is the data transfer itself. The sales person will wax poetic on the data acceptance capabilities of his package. This is great, but is all the necessary data available in your present system?

The other item to consider is the master files. Are you sure they can be kept in sync? There must be capabilities to do this. You should ask who else is interfacing with your package.

Configurable - The word itself means that there are options on the way the system will handle certain conditions. It also can be a source of extra revenue for the vendor. When the sales person uses this word, ask, "by whom?" Some of these configuration options are permanent and others may be changed.

Permanent options (note the word "option") must be given more thought than the ones that can be changed. Be sure that everyone who will be making these decisions knows which ones are permanent and which may be changed. You should always know who makes the changes.

If you are to make the changes, you should know how often they can be or must be changed. You also should examine the documentation on them. It could be that technically you could make the changes but no one could understand the documentation. Therefore, the vendor makes the money.

Performance (Power) - Do not believe what you hear or read. These are entirely relative terms and have little basis in fact. The only way to compare one software system's performance to another's is to put them on the same machine. Give them the same data to process one after the other. The machine also must be unloaded.

If you cannot do the test described above, there are alternatives. The most financially attractive is through reference contact. Ask companies with similar transaction volumes and machines about timing. It is not as accurate as direct comparison, but it is better than crunching theoretical numbers.

___-like - What are "English-like commands?" German and Latin contain some words that sound like English, so are these "English-like?" The same concept is used in software sales. The vendor will claim that his software has commands that use the same context as a commonly used package or utility. This leads you to believe that everyone who uses package "A" will be able to use package "B" immediately. It may not be the case.

The best way to determine who will be able to use the package is to let the target group use it for a while. See how fast they can learn it. If most of them learn it quickly, the rest will follow and you will have a usable package. If few of them learn it, find something else.

We're working on that - Whenever a good idea comes up in a demonstration and it is in the software, you will hear this phrase. You are supposed to imagine hundreds of programmers busily putting this into the software. It could be that the sales person has just started working on it now.
The author of this paper is working on being a millionaire. He is not doing a great job at it, but he is working on it. This should be your attitude when you hear this phrase. How hard is the software company working on it? What is its state?

Bad Demo Data - When unexpected results occur during the demonstration, this is the sales person's battle cry. The problem with bad demo data is that it may show a lack of control on the vendor's part. Software sales are the most important part of the vendor's business. If they cannot control this, what can they control?

Another problem also exists. The sales person may be covering a bug with this excuse. All software has bugs. The nature of the bugs and the way they are dealt with is important. If you were asking to see something unusual and it blew up, this may be a bug. If the sales person attempts to cover it up, it may be a problem.

We have ___ packages installed world wide - Software companies want to appear as big as possible. They will inflate the numbers with words like packages instead of customers. A package can be a system such as GL or AP or it could be a module such as a report package. It would therefore be possible for a software company to claim 500 packages with only 20 customers.

Another trick is to count multiple sites with multiple "packages." This means that a customer with three sites, three packages each (GL, report package and a PC module) would count as nine. If you can get down to the number of customers, get a complete list. You should compare the number the sales person quoted to the number on the list. He should reconcile any differences.

We are talking (working) with an organization that - This entity does *exactly* what you are trying to do. This is too convenient. It is much too convenient. The sales person might have met someone in a bar three nights ago that works as a janitor in the same type of organization as yours. The sales person is not lying, he is talking with such an entity.

You can prevent a misunderstanding (misstatement) of the facts through follow up. Ask the sales person for the name of the organization and a contact. If he will not give you one, discount the story as fiction.

Complete Consulting Services - Such a deal, one stop shopping! This sounds better than it may be. The vendor may be depending upon the consulting side for revenue. This can result in many problems.

The first problem is called the "monopoly syndrome," and it is no game. The vendor will attempt to control all consulting and training on its package. If it is successful in obtaining this monopoly, the price of these services is at the vendor's mercy. Another problem is that your project's timing also may be subject to the whim of the vendor. There is a way to discover if the vendor has this problem.

One way to find this out is to ask the sales person if there are alternative sources of training or other help on the package. If he answers yes, then find out who they are. Any other answer may show an attempt by the vendor to monopolize service on the package.

Another way to find other sources is to ask other customers. They will tell you where to look. This information (not the customer name) should then be given to the person in charge of support at the vendor. The reaction can suggest whether the vendor approves of outside consultants or not.

It should be noted that most vendors want to do their own basic training to ensure it is done correctly. Outside help is generally needed for areas above and beyond basic training.

Another problem with vendors who depend upon consulting revenue comes when determining what service charges should be. These vendors will be more inclined to charge customers for small services. The charges are sometimes excessive. It is difficult to determine where free service ends and charging begins.

It is easy to find this type of vendor. Comments from customers like, "they tend to nickel and dime you" or "there were more charges than I thought," will give them away. If you do not get these comments, ask the customer if there are a lot of charges for service. A yes answer suggests caution.

Response time - This is a subset of performance. There is no adequate way to determine response time without using your data on your machine. There is a trick you can use to determine which areas are weakest in a given piece of software.

The trick is to watch the sales person. If he moves the keys a while, pauses, talks and then hits the enter key, watch out. He may have hit the enter key during the previous flurry and is now covering this up. If you suspect this, have him log off and redo the steps. You should watch carefully.

Summary

The sales person will use several of these words and phrases during conversations and demonstrations. They are carefully chosen for both annotative and connotative effect. Listen to each word and phrase that the sales person uses. Be aware of these terms and note the alternative meaning of each. Follow up any suspicions with other customers. In short, do your homework.

Paper#: 2006

Title: Ready ... Shoot ... Aim
Management Perceptions of 4GL Tools

Author Name: Donald J. Hill

Company: Comprehensive Software Solutions
300 N. State, Suite 3125
Chicago, IL. 60610
(312) 222-0758

OBJECTIVE

The objective of this paper is to examine the differing *perceptions* of MIS versus Non-MIS organization members vis-a-vis Fourth Generation Language and CASE tools (4GL/CASE). Ready ... Shoot ... Aim will demonstrate that different perceptions, or more precisely, paradigms can markedly effect 4GL/CASE utilization and effectiveness.

The concept of paradigms has been explored recently in business literature. Stephen R. Covey in *The 7 Habits of Highly Effective People* describes paradigms as:

> ... The word *paradigm* comes from the Greek. It was originally a scientific term, and is more commonly used today to mean a model, theory, perception, assumption, or frame of reference. In the more general sense, it's the way we 'see' the world -- not in terms of our visual sense of sight, but in terms of perceiving, understanding, or interpreting. ... For our purposes, a simple way to understand paradigms is to see them as maps. We all know that 'the map is not the territory.' A map is simply an explanation of certain aspects of the territory. That's exactly what a paradigm is. It is a theory, an explanation, or model of something else.

Our consulting firm has a very committed 4GL/CASE paradigm -- we fully utilize 4GL efficiencies and CASE methodologies to bring the system's life cycle to fruition. Many articles, papers, books, and in some cases 'religions' have evolved around the 4GL tools and CASE methodologies. This paper will assume the benefits of 4GL/CASE. The benefits and explanations of methodologies are technical in nature. This paper will explore the non-technical (and seldom written about) aspects: How differing MIS vs. Non-MIS paradigms can effect 4GL effectiveness and utilization.

4GL/CASE EVOLUTION

In order to understand the paradigms evolving from Fourth Generation Languages (4GL's) and CASE tools, one needs to explore the other generations of programming languages.

The first generation (1GL) was a 'programming' language intended for scientists. The computers used were the size of a large room and could only be programmed by re-wiring to accomplish programming changes. The perception: Since there was no MIS department, all users had the paradigm of seeing little business application for the new computers.

The second generation (2GL) was a 'programming' language intended for engineers. The computers were a little bit smaller and quite a bit faster, but could only be programmed by machine language. You had to be a 'rocket scientist' to program in this 'language'. The perception: As it was becoming a little easier to program the still new computer, some potential business applications were devised. The MIS department (then called Data Processing) were seen as wizards by the rest of the organization.

The third generation (3GL) was a programming language intended for programmers. Computers were becoming cheaper and faster, and more data needed processing. Now programmers could operate in quasi-English computer languages and code programs in a fraction of the time of 2GL's. The perception: The non-MIS organization looked upon MIS as technicians that would translate their business needs into the somewhat arcane computer languages of COBOL, FORTRAN, etc.

The fourth generation (4GL) was originally intended as a programming language intended for the end-users in an organization. Computers had become bigger, faster, and cheaper. More members of the organization demanded (and in most cases needed) more timely access to information. In James Martin's revolutionary book *Applications Development Without Programmers*, the world was seen as users doing their own applications while using English like commands in a 4GL. Information Centers became very popular and very effective. To summarize, the main difference between 3GL's and 4GL's is that both languages tell the computer what to do -- a 4GL does not need to specify how to do it. The perception: MIS had lost some of the mystique (i.e. this computer stuff isn't all that hard, is it). Non-MIS organization could tap the power of the computer without MIS liaison (or control). Through the advent of PC's and end-user oriented 4GL's, many more people could access utilize a computerized information. Many end-users were computer literate yet didn't understand the intricacies of system design and integration. They were computer literate but not systems literate.

The fourth generation evolved in the MIS area also. As the technology of 4GL's began to catch up with the technology of the computers themselves, programmers began utilizing these new tools. 4GL vendors began to actually sell 4GL's directly to MIS departments as productivity tools for programmers. Claims of 10 - 100 times improvement in productivity were substantiated -- one could indeed CODE 10 to 100 times faster in a 4GL. It became very attractive to code programs after little or no systems analysis and design. Ready .. Shoot .. Aim was born.

Finally, and in some cases a backlash of the Ready .. Shoot .. Aim syndrome, the emergence of CASE (Computer Assisted Software Engineering) occurred. The common thread of the many methodologies is ANALYSIS. A very good analogy in Alan Fisher's *CASE: Using Software Development Tools* is how confident would you be in a house if your developer just started building -- and then proceeded to draw the sketches and blueprints. The perception of Non-MIS organization members in some cases: We don't have time to do the sketches and blueprints -- we need the house done tomorrow. Also, if the corporate culture is 'shoot from the hip', it can be even more difficult to Ready ... Aim ... and then Shoot. Yet, even in corporate cultures of not shooting from the hip, Ready .. Shoot .. Aim can occur. Carefully laid out business plans, forecasting, modeling, etc. do not necessarily imply careful systems analysis and design in the MIS organization.

Before we can analyze the syndrome, we need to discuss exactly what Ready .. Shoot .. Aim is.

THE READY .. SHOOT .. AIM SYNDROME

Ready ... Shoot ... Aim is a syndrome that results from an incorrect paradigm vis-a-vis 4GL/CASE tools. The analogy that will be used is a rather simple one. We have a target -- a correctly completed computer application. Our goal is place a number of well placed bullets in the center rings of the target to consider the application successfully completed. We have a fourth generation language (4GL) -- an automatic rifle to help us shoot fast and straight. We have a CASE tool -- a telescopic sight to help us aim correctly. The final rules of the analogy: There is a limited supply of ammunition and fixed amount of time to accomplish our goal of hitting the target.

The components of the 'correct' paradigm are Ready, Aim, and Shoot. READY involves a needs analysis, cost benefit, payback, and resources involved in a proposed system. In addition, future implications need to be addressed. AIM involves system analysis and design, data modeling, user involvement, timelines, priorities, etc. Maintenance design and planning is an AIM activity. SHOOT involves coding, testing, implementation -- and maintenance. Ready, Aim, and Shoot are the elements of the systems life cycle. The role of CASE tools is to clearly define inputs from the Ready stage and streamline the Aim stage. A 4GL can provide dramatic productivity gains in the Shoot stage. While it appears intuitively logical to follow the life cycle steps, this progression does not always happen. Why?

Sometimes the natural progression of the systems life cycle cannot be expedited due to technical reasons. Those issues will not be addressed here (in fact a 'quick fix' can be implemented by simply choosing a more appropriate methodology or more effective 4GL). However, in our experience as consultants, we find that perceptions are usually the culprit in the systems life cycle breakdown. So just change the perceptions, right? Unfortunately, changing perceptions (or to coin a 90's term - paradigm shift) is not a trivial matter. Before one can affect a paradigm shift, one must understand the existing perceptions ... the Ready .. Shoot .. Aim syndrome.

THE SYNDROME

The landmark book *Applications Development Without Programmers* was mentioned earlier as being the introduction of the 4GL mentality. If we were to write a book about the Ready .. Shoot .. Aim methodology, it would be 'Applications Development Without Analysts'. Why is this backwards life cycle methodology prevalent ? Returning to our analogy -- the automatic rifle (4GL) is sitting there. The clock is ticking. We have a target in mind (we can't see it very well but it is in mind). We want to hear some noise. We know that we should carefully define the target, use our CASE tool (our telescopic sight), adjust it, zero in on the target ..... But what fun is that. It is terribly alluring to pick up that rifle and let the shooting begin. Our company paradigm defines the fun as being the aim phase.

Recently, we completed our report normalization workshop (a methodology we've developed specifically for interpreting user input and translating the request into a specification) -- one of the participants stated "You've taken all of the mystery out of programming". Obviously, that person's paradigm was different than mine (I don't like mystery novels, let alone mystery in the final stages of the systems life cycle). Yet, what if the corporate culture is shooting from the hip and our boss issues the edict: "The enemy is getting closer. Why aren't we shooting. Let's here some noise !". You calmly explain the benefits of careful analysis and planning and state that we have a much better chance of hitting the target if we, uh, aim first ? (The technical term: I am trying to get my boss to undertake a paradigm shift). Your boss replies "I agree with you 100%, but while you were talking, the enemy is even closer than before ... I want to hear some NOISE ! ! !". Not being independently wealthy, you start shooting. And yet, because you have a 4GL, you can code the system very quickly. But is it the right system ? The point will not be belabored nor will war stories be told of the disasters that follow from Ready .. Shoot .. Aim. But the example does beg the question: Human nature being what it is, didn't people always want to "hear some noise now !".

Actually, yes. However, if 4GL's are likened to automatic rifles, 2GL's and 3GL's can be likened to flint lock guns. One had to load the ball into the barrel, put in the powder, pack it carefully, etc. before one could shoot. Much more care (AIMing) had to be exercised for one did not have much of an opportunity to shoot again quickly in case of a miss. Also, in the old days of 2/3GL programming, computers had less resources and were slower -- programs had to be designed to be very efficient. In addition, fewer non-MIS organization members could utilize the information directly or had computer literacy on PC's. So, Ready .. Shoot .. Aim was not always possible in the 'old days' -- today the syndrome is much more alluring. 'Shoot from the hip' corporate cultures are more readily transferred to MIS environments. Simply stated, the technology (new computers and better 4GL tools) do allow, if not promote, the syndrome.

A question that arises is: "Can Ready .. Shoot .. Aim actually work ?." If this methodology is the only available means of achieving the systems life cycle, it will work as long as ammunition is available or users 'settle' for whatever is implemented. It will also work if the corporate culture is attuned to acting quickly and accepting the results of inadequate analysis and design.

The methodology can also work if you are lucky. In *An Introduction to Information Engineering*, the concept of Ready .. Shoot .. Aim was likened to utilizing a 4GL to code the wrong system faster. The idea here is code more systems fast enough and hit the right solution in less time than it would take to do the appropriate analysis and design. The problem is that one cannot quantify how many wrong systems it will take before the correct one is stumbled upon. Echoing the question asked in *Dirty Harry* -- "Well organization, do you feel lucky". Assuming that we don't feel lucky and that our goal IS in fact Ready .. Aim .. Shoot, we need to understand the perceptions that can lead us to Ready .. Shoot .. Aim.

In order to understand the perceptions, it is often useful to examine the paradigms vis-a-vis who proposes or 'owns' the 4GL decision in an organization -- and the relation with the organization as a whole. This exercise is certainly a useful tool in understanding the 4GL perceptions, but it also can be used to analyze virtually any situation or problem. Much meaning can be gained by understanding and defining your own paradigms, as well as gaining the understanding of other's perceptions.

Proposer / Owner <-> User Paradigm Matrix

Possible Paradigms

Proposer	MIS	Non-MIS
Non-MIS		
MIS		
4GL Vendor		

First, all possible scenarios will not be described. What is important is the exercise of understanding 'our' paradigms and 'their' paradigms. The key is empathy: Putting aside your own paradigms for the moment and analyzing 'their' perceptions.

A typical scenario is the 4GL vendor acting as the proposer. Often the resulting paradigm for MIS is reducing applications backlog. More systems can be churned out in a quicker amount of time. The paradigm for non-MIS management (who approves the purchase of the 4GL) is a 10-100 fold increase in productivity. In such situations, the perceptions do not reflect that such an increase in productivity occurs in coding and testing, typically only 30 - 40% of the applications life cycle. Remember, the 4GL is a SHOOT tool. READY and AIM are not addressed by 4GL's. Interestingly, MIS members often fall into the same trap.

Another scenario is an 'MIS Driven' selection, approval, and budget decision. MIS paradigms may reflect the importance of the SHOOT tool -- we can shoot better and faster. Or, the perception could be that the 4GL is an AIM tool. While a 4GL can be utilized very effectively as a prototype vehicle in the AIM phase of the systems life cycle, often prototyping is a substitute for vital analysis and design functions. The perceptions of Non-MIS organization members can be varied -- a Ready .. Shoot .. Aim paradigm can occur here when users a computer literate vis-a-vis PC productivity tools but are not systems literate (the mindset of "I could do this whole thing on my PC in a day"). 4GL programmers or end-users acting as programmers do not necessarily make good analysts or system designers. In addition, the syndrome could be promulgated if the 4GL is seen as a panacea for organizational difficulties.

The final scenario that will be explored is the Non-MIS driven 4GL decision (with the possibility of MIS being 'out of the loop'). This scenario usually occurs in small MIS organizations and potentially is the most difficult obstacle to obtaining Ready .. Aim .. Shoot. The paradigm of Non-MIS organization members here is the 10-100 fold increase in productivity in the systems life cycle (not just in the coding and testing area). They are absolutely amazed at the 4GL demonstrations and have the perception "Push a few buttons and we have our system". In most cases, a large investment has been made in the 4GL and time spent in analysis and design is considered wasted. MIS is in a very difficult situation regardless of their paradigm.

In summary, it is vital to fully understand what the paradigms (both yours and theirs) is before on can utilize any 4GL tool effectively.

OTHER SYNDROMES THAT CAUSE READY .. SHOOT .. AIM

After the perceptions are empathized, it would appear to be not that difficult to accomplish a paradigm shift. Many times those paradigms are based on past experience with MIS or from past experience. In other words, other symptoms can exist that cause Ready .. Shoot .. Aim.

Ready .. Aim .. Aim .. Aim .. Never Shoot: This syndrome implies that so much analysis (AIMing) was done, we never had a chance to shoot at the target. The syndrome can point to a lack (or an absence) of work in the READY stage (i.e. we never really understood what are target was). An excellent 'fix' to this situation is utilizing data modeling and information engineering -- the users define business goals and models and those models actually drive the analysis and design. In addition, a lack of or non-adherence to proper analysis methodologies and/or CASE tools could also be a cause of this syndrome. In any case, this syndrome promotes much impatience and almost forces results quickly -- even if they are not the right results.

Ready .. Aim .. Shoot .. Miss .. Aim .. Shoot .. Miss ...: This syndrome implies, of course, faulty aim. In our analogy, our telescopic site is blurred. Again, we could the exact same causes as above -- and the usual fix should be a better analysis and design methodology. However, the result is to usually just try to shoot faster.

Ready .. Shoot .. Aim .. Miss .. Shoot .. Aim .. Miss ...: This syndrome is slightly better than the above scenario. One is actually stopping to examine the results of the missed shot before taking another. What could very well be happening here is prototyping -- prototyping used as a SHOOT tool versus an AIM tool. The obvious fix is to incorporate the systems prototyping and utilize it correctly in the analysis and design phase of the life cycle. However, the frustration level of the eventual system users can become quite high. Usually this situation regresses to Shoot .. Miss .. Shoot .. Miss.

Shoot .. Shoot .. Shoot .. Shoot: While this syndrome appears ludicrous (we can't just start shooting, can we), situations do occur where there is no READY, nor is there any AIM. This is the quintessential 'shoot from the hip' organization. The symptom is a general lack of analysis and planning -- we want something to happen -- anything. What will usually happen here is a progression (not regression) to Ready .. Shoot .. Aim.

Again, we want to achieve Ready .. Aim .. Shoot. The perceptions have been examined, as well as several other syndromes that allow the Ready .. Shoot .. Aim paradigm to be practiced. By using these inputs, one can attempt to consider...

ACHIEVING READY .. AIM .. SHOOT

Our situation: We are experiencing one of the above symptoms or variations or we are about to undertake a major project and would like to achieve Ready .. Aim .. Shoot. First, do we have adequate READY tools -- a sound business plan, system survey, needs analysis, etc. Do have a target and a means of hitting that target. Is it important that we in fact hit the target ? Secondly, do we have adequate AIM tools -- a sound analysis and design methodology and/or CASE tools. Have we clearly defined the target and how we will hit that target with the minimum amount of ammunition. Finally, do we have adequate SHOOT tools -- a robust and flexible 4GL. Can we shoot rapidly and accurately at that target. If all of these are in fact in place, we can proceed to our paradigms and other's paradigms.

Before we can pursue a Ready .. Aim .. Shoot strategy, paradigms supporting the strategy must be in place. Again, we should return to our earlier 'us/them' paradigm matrix and understand the perceptions. In short, before we can accomplish a paradigm shift, we need to understand what the existing perceptions are. The tool for this understanding is empathy. A very apt description appears in Covey's *The 7 Habits of Highly Effective People* chapter entitled "Seek first to understand, then to be understood". Finally, an element of trust must exist. Good systems analysis and design WILL provide the right system faster ... we DO have time to do it the right way.

Sometimes a compromise can be struck with Ready .. Aim .. Shoot -- some paradigms do not get shifted easily. Actually, our consulting firm utilizes a compromise to Ready .. Aim .. Shoot. Often, if not always, we are measured by producing demonstrable results quickly. In all modesty, we are better 'shooters' than most. However, we cannot resort to Ready .. Shoot .. Aim -- we're not that good (or lucky). We have developed a methodology that is very consistent with methodologies such as Yourdon's, etc. and works quite well.

We practice Ready .. Aim .. Shoot .. Aim .. Shoot .. Aim .. Shoot -- until the objective is met. We break apart (normalize) the entire project into stand-alone modules. We can then work on them concurrently if needed. Simply, we are bypassing the top-down approach of other methodologies. Again, this can be effective even in 'shoot from the hip' corporate cultures.

SUMMARY / CONCLUSIONS

One could infer that this paper is anti-4GL (many of the analysis and design methodology books are). This is certainly not the intent. In fact, our organization is a preferred SPEEDWARE consulting firm -- we owe much of our success to the use of that 4GL. Much of that success involves utilizing the 4GL correctly -- as a SHOOT tool -- and carefully utilizing its prototyping capabilities as an AIM tool. We also would not even consider attempting a system without adequate systems analysis and design. To complete our analogy, we define our target and what we are going to use to hit the target. If we decide our automatic rifle (4GL) is appropriate to hit the target, we take careful aim with our telescopic sight (our analysis/design methodology and CASE tools). Only then are we ready to shoot. We calmly, without panic, place our shots into the middle rings of the target. The objectives are met and nobody is struck by stray bullets (friendly fire). Ready .. Aim .. Shoot has worked.

The right tools and an understanding of the paradigms involved will allow Ready .. Shoot .. Aim or a feasible alternative.

John Podkomorski
Hewlett Packard Company
1266 Kifer Road
Sunnyvale, CA 94086

Information Technology and Management Information System (IT/MIS) management is becoming more complex as a direct result of industry forces, technology, and the changing expectations of management, users, and staff. These changes are often subtle, but have a very significant, and often confusing, effect on the day to day management requirements of the IT/MIS management team.

This session develops a model to help understand the changes which affect the IT/MIS environment. Historical pressures will be compared to new pressures for the decade of the 1990's. In addition, a group discussion will focus on possible strategies for managing the new set of pressures felt by IT/MIS managers.

AUDIENCE:

All levels of IT/MIS management, especially department management, and the management and staff of application development , systems design and implementation, and technology support.

A TIME OF RAPID CHANGE

The IT/MIS management world is becoming much more complex. Change has become so very rapid that it often feels as if the comfortable patterns of the past are all changing at the same time.

Where will IT/MIS managers feel pressure through the 1990's? How are the pressures changing? And why?

There are two basic sources of increased pressure on IT/MIS Management:

1. Technological development, and industry trends.
2. Organizational pressures from user communities inside each company.
   A. Upper Management
   B. Users of IT/MIS services
   C. IT/MIS Staff

This paper will discuss the changes in each of these very important areas.

This model is not really very creative. These factors have been the major driving force behind IT/MIS for many years. What is changing rapidly, however, is the effect of industry developments and technology progress, and as we'll see later, the set of expectations each of the three organizational "users" now have for IT/MIS.

First, let's look at the effect of industry and technology changes.

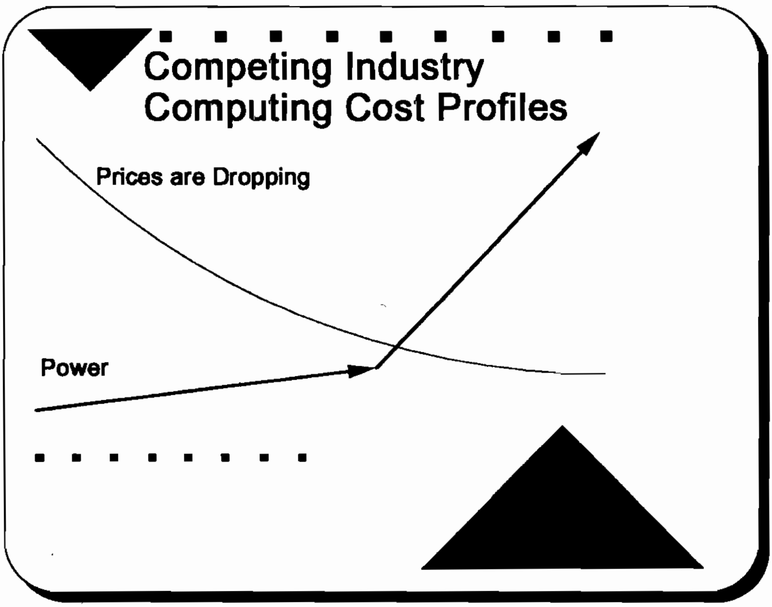
This is the area which is the strongest contributor to the rapid change in the IT/MIS environment. Technology change is happening at an unprecedented pace, throughout the industry. At the same time that technology is changing rapidly, we've seen that the forces of business changes are moving equally rapidly, with unprecedented effects on competitive pricing. The combination of these two factors amplifies the effects of both.

Let's look at just a few of the industry issues IT/MIS needs to deal with.

PRICES ARE DROPPING

Computer price performance curves are changing dramatically. Some time ago, a computer vendor company was considered aggressively successful if it could maintain a 20% per year improvement in price performance. New computer models were released to market perhaps once every two years, and a 5 - 8 year expected life cycle was not uncommon. There was a clearly visible difference between the "power" of a mini-computer, and the "power" of a mainframe.

PERFORMANCE IS INCREASING DRAMATICALLY



As a direct result of RISC Architectures, today's vendors are increasing the power in their processors at nearly 200% per year. Just 4 years ago this rate of increase was respectable at 20% per year. It is not unusual to see very rapid announcements of new computer lines which totally replace the "state of the art" just 9-12 months back.

The historical gap between "mini-computers" and "mainframes" is not as clear today In today's market, there really are alternatives to the mainframes of the past. MIPS, and very large disk and network configurations, are readily available from very high power processors, often fully software compatible with the distributed processors currently in place. "Mini-computers" supporting many hundreds, and soon to be thousands, of users are common. The management of these computers requires different resource algorithms than those used on minicomputers in the recent past.

One expectation which has not changed, however, is that the operation of these more powerful computers will be as easy, or preferably easier, than the operation of previously distributed mini-computers, and will certainly never be as complex or expensive as managing a mainframe environment.

PURCHASE DECISIONS ARE HARDER

Purchase decisions are increasingly difficult as a result. IT/MIS must ask two questions, and balance the answers carefully...

> What if I buy today, and a newer, more powerful, cheaper model is released in 3 months? I'll really look silly!!!
>
> What if I don't buy today, and my need for increased computing causes a business problem for my company? I'll really look silly!!!.

ECONOMY OF SCALE

In an interesting turn of events, the significant increase in the power of mini-computers is causing many companies to rethink their decisions to "decentralize". In very large corporations, there is an increasing trend toward increasing the size of computing facilities, with a strong emphasis on economy of scale and improved IT/MIS operational efficiency.

This trend, however, is quite different from a return to mainframes. Companies really want the ease of use and operational simplicity of a mini-computer environment, while also getting the advantage of economies of scale in larger operations.

SOME SIDE EFFECTS:

There are several side effects caused by larger organizations, including financial factors, new management skills, new vertically integrated work groups , and a significantly higher dependance on data communications technology.

FINANCIAL FACTORS:

Historically, many, if not most, decisions in IT/MIS were left to the technical discretion of the IT/MIS manager. Computing technology was not commonly understood by management, and as a result the research, judgment, and decision making was left largely to the IT/MIS community. Distributed operations in the past several years were justified by their immediate customers, and accounting, or chargeback systems, were not important.

This is changing today, due mostly to the increasing pressure on financial performance, and considerably higher attention by "upper" management. As more technologically trained managers rise through the ranks, financial pressure is often reflected onto technical decisions. The chargeback systems come under close scrutiny. Service level agreements need to be negotiated between managers.

IT/ MIS managers no longer get to say "Trust me - its a technical thing!!".

NEW MANAGEMENT SKILLS:

Minicomputer IT/MIS organizations of the recent past have been proportionately small, especially when compared to "typical" mainframe shops. Performance or capacity planning was proportionately simple, with many processors managing a single workload, or a limited combination of focussed workloads. As more powerful computers are implemented, IT/MIS management must implement many of the techniques formerly only used on mainframes. Increased capacity planning and forecasting is critical. Accurate and representative chargeback systems are suddenly necessary. IT/MIS is evolving from a technical service to a "business function".

Also, the customers of IT/MIS are becoming more computer literate, and as a result are demanding more comprehensive and comprehensible data about the operations of their computers. Performance and billing metrics are increasingly necessary, stated in business terms which are understandable by end users.

VERTICAL WORK GROUPS:

There seems to be a current trend in industry emphasizing flatter organization charts with fewer layers of management. This style of organization has a much different decision making model, especially compared to a deeper, more monolithic structure. There is a higher need for rapid, effective, and efficient information dissemination, as well as a requirement for flexibility, and responsiveness to rapid change.

In smaller, often distributed, organizations it is common to have a very "general" work force, with fewer employees managing a processor, each with a very broad set of skills. As organizations get larger, it is increasingly necessary to develop technology specialists, with less breadth in their assignments, and considerably more depth.

DATA COMMUNICATIONS:

Larger processors also require a higher dependence on data communications capabilities. Some of the benefits of decentralization included a user's sense of control, with access at the user's desk, and listings and reports delivered quickly very close to the user's work place. As processors get larger, distances from users also get larger, and the need to distribute processing power (PC's, workstations, terminals), as well as remote printouts and reports, depends on data communications.

CHANGING EXPECTATIONS

IT/MIS managers will need to change their expectations, and develop new skills to match the evolution of the industry. IT/MIS managers need to increase their attention to the business factors in running their shops, aligning with the goals of their company, and closely accounting for the resources used.

There seems to be a current trend in industry emphasizing flatter organization charts with fewer layers of management. This style of organization has a much different decision making model, especially compared to a deeper, more monolithic structure. There is a higher need for rapid, effective, and efficient information dissemination, as well as a requirement for flexibility, and responsiveness to rapid change.

CHANGING VENDOR / CUSTOMER RELATIONSHIPS

We've seen earlier that prices for computing equipment are dropping rapidly, and that the power delivered by this less expensive equipment is increasing rapidly, resulting in extra-ordinary price performance for purchasers of computing power.

Unfortunately, the rapidly lowering prices are causing considerable stress in the funding algorithms used by most vendors. From a customer's perspective, things which once were "free" seem to be no longer "free". Most vendors are being forced into finding more efficient, less personal processes to sell their equipment, and no longer have the freedom to include many of the "extras" we've come to know and love.

STANDARDS

And then there is the "open systems" question. What is the "right" thing to do in an IT/MIS department? How do we decide whether to make changes across vendor boundaries, or Operating System boundaries? What criteria should we use? If we stay proprietary, will that mean we're "trapped"?

Can your company afford to do the development necessary to re-implement all applications on an open systems platform? Can your company afford retaining the status quo?

Will our staff continue to work for us if we don't go to UNIX? Will our staff continue to work for us if we do go to UNIX?

APPLICATION DEVELOPMENT TECHNOLOGY

Another contributing factor is the development of new technology in application development methodology.

The recent entry of computer engineering (CASE) products into the marketplace is changing the expectations of IT/MIS staffs, as well as those of our users. CASE tools are reported to make development much faster, and with "perfect" results. Developers and designers are beginning to demand access to CASE technology (perhaps in self-defense) in order to increase their own productivity.

4GL's have also increased our user's expectations about the speed with which applications can be delivered. 4GL's promise development times considerably lower than historical methods.

4GL's have another interesting effect. As computer literacy increases in the user base, more and more users are feeling competent in inventing or modifying their own applications. IT/MIS has considerably less control in an environment where users can write inquiries, reports, and even update their own records using 4GL technology.

The advent of relational data bases, and SQL access methods, amplifies this trend. As the elements of a company's data become easier to manage, and more intuitive to access, the nature of system design and implementation must change to allow for the considerably higher level of user freedom currently available. "Object" orientation is challenging the technology limits again.

PC INTEGRATION

Increased power in PC's and workstations, coupled with the software technology to utilize that power for user interface enhancements, and for file and data storage distribution is really changing the profiles of what new applications look like. The definition of "central" computing, or "distributed" computing, is getting pretty fuzzy.

Also, development of software packages - "shrink wrapped" software - is increasing our users' appetite for solutions. From a user's perspective, the software and applications marketplace infers that every business problem has a solution "on the shelf". All a company has to do is buy it, and all their problems will be solved. In some cases, that's true. In others, it is not very accurate.

Tough decisions everywhere...

Which is why you experience all the HYPE!!

We vendors are a very creative lot, and will find many ways to present each technical idea, product, or fantasy in a way which makes it sound like the very best thing that ever happened to the world.

HISTORICAL MANAGEMENT EXPECTATIONS

Let's look at the evolving expectations of Management. By management we mean the organization structure to which IT/MIS reports. Historically, this has been the Finance function in most companies, or perhaps the Manufacturing group, depending on the needs of each company. One change we see developing is the continued movement toward stand alone IT/MIS functions, at a par with other functions in the company. This change is one of the forces driving new sets of expectations.

AVAILABLE, BUT INVISIBLE

Historically, IT/MIS departments have been expected to be invisible, and have had the luxury of working in a limited monopoly situation. They had the technical experts, the equipment, the technology knowledge, and the endless jargon which kept the unruly users at bay.

Management expected the department to operate without complaints or surprises, and to manage their budgets with reasonable controls in place.

This expectation set has not changed significantly in the last 20 years or so.

NEW EXPECTATIONS

What *has* changed?

NEW MEMBERSHIP IN THE MANAGEMENT TEAM

The most significant change, which is made more visible by the movement to have separate IT/MIS functions in large companies, is the expectation that IT/MIS become more of a business partner with the other functions in the company. New demands are drawing MIS managers more into the management team to provide the "systems" expertise to solve business problems.
Current financial market conditions are forcing many companies to redesign business processes. IT/MIS organizations frequently have the professional skills to drive that redesign, and are increasingly being asked to do so.

Information management is increasingly expected to create a competitive advantage - to make a contribution to company revenue. Increased productivity, improved customer satisfaction, faster deliveries - all these contribute directly to company bottom lines, and all can be improved by more user-friendly and productive applications systems.

Instead of a "servant" to the management team, IT/MIS is now being asked to be an active partner, with considerable emphasis on the IT/MIS contribution to organizational development, profitability, and business process improvement. IT/MIS authority seems to be increasing, with an absolutely necessary shift away from the comfortable ground of technology to the much more slippery arena of business management. This often brings along an increase in authority for IT/MIS management, but not necessarily with the enthusiastic endorsement of the rest of the management team. The "politics" have changed.

Recent press articles, however, have begun to suggest that upper management is not satisfied with current MIS contributions.

EMPHASIS ON FINANCES INSTEAD OF TECHNOLOGY

Another change in the dynamics of companies is the increased assertiveness of the finance function over technical decisions once tightly held in IT/MIS.

Because of increased technical exposure, many more managers are becoming remarkably competent at asking very hard questions about the economics of IT/MIS development and operations. We have seen real cases where a change as radical as a shift in vendors has been caused by a directed budgetary limitation.

> "If finance says we must operate for $1M less per year, how can we do that? We may need to switch vendors!!".

GLOBAL REQUIREMENTS

Global influence is rapidly rising. Companies can be "national", "bi-national", "multi-national", or "global". Because of the shrinking size of the world, many more companies are seeking opportunities to enter the global marketplace. The world market brings along many very interesting management challenges, and a whole plethora of technology challenges, which were not commonly considered even 10 years ago.

U.S. companies, in particular, are unaccustomed to the stresses of international or global business. This increasing "global" requirement represents a new type of thought. Cultural issues, as well as the complexity of international finance, standards, and inter-country laws must be considered as an active part of "system implementation" decisions.

Here are some results from a recent survey done by the INDEX GROUP for 1991. The survey participants were a group of 394 U.S. and Canadian I/S executives, with companies greater than $250 million. 74% of these companies had revenues over $1 billion.

The chart includes the results from the last four years. It is very interesting to look at the rate of change in many of the items, as well as the number of items which were not on the list even two years ago, and have risen greatly in importance.

Let me draw your attention to some specific items.

#1 on the list is "Reshaping Business Processes through Information Technology". In 1988, this was not thought to be a major issue. For the last two years, however, it has been #1.

#10 on the list, "Enhancing Leadership", has not been mentioned before. In 1991, however, it has risen to #10. This is very key to the success of IT/MIS organizations. Upper management expects much higher leadership than it has in the past. At the same time, upper management is not sure that current IT/MIS management has the leadership skills to deliver against those expectations.

#4, and #7 on the list specifically address 1991's emphasis on the highly productive development of high quality software. It is easy to see the increased emphasis placed on productivity and quality.

1991 **INDEX GROUP** SURVEY RESULTS

1988 Rank	1989 Rank	1990 Rank	1991 Rank	DESCRIPTION
None	11	1	**=>1**	**Reshaping Business Processes through Information Technology**
1	2	4	2	Aligning Information Systems with Corporate Goals
None	7	3	3	Instituting Cross Functional Systems
12	13	6	**=>4**	**Boosting S/W Development Productivity**
7	6	7	5	Utilizing Data so that information is accessible and used by the right person at the right time
2	4	5	6	Developing an Information Systems Strategic Plan
None	None	14	**=>7**	**Improving Software Development Quality**
5	5	9	8	Creating an Information Systems Architecture
6	12	16	9	Integrating Information systems, solving technical problems of integrating heterogeneous h/w and s/w
None	None	None	**=>10**	**Enhancing Leadership Skills at or near the top of the IT/MIS Organization**
17	14	10	11	Cutting IT/MIS Costs
4	1	8	12	Using IT/MIS for Competitive Breakthroughs
8	8	11	13	Improving the IT/MIS human resource
3	3	2	14	Educating Management on IT/MIS
None	None	19	15	Connecting to Customers and Suppliers
9	10	17	16	Managing Changes caused by IT/MIS
None	None	15	17	Promoting the IT/MIS Function
None	20	23	18	Determining the value of IT/MIS
13	16	25	19	Managing dispersed Systems
None	17	21	20	Capitalizing on advances in Technology

As you can see, the list of top 20 needs of the IT/MIS have shifted a little over the last four years, with management needs expanding, and technology needs dropping a bit.

Let's shift, now, and look at how the expectations of USERS of IT/MIS are changing.

HISTORICAL USER EXPECTATIONS

"Users", in this context, are the departments in a company who actually receive the benefit of IT/MIS development and operations work.

Historically, users have expected solid, reliable, available, cost effective services from IT/MIS. This, too, has not really changed in 20 years. Users have always been concerned about their application development backlog, but have grudgingly accepted it because of the perceived difficulty of implementation of complex IT/MIS programming projects.

In historical IT/MIS installations, user personnel would operate a single application, often heads-down to the keyboard, for hours on end. They were specially trained for that application, and worked with it for a long time. Specific application functions needed to be well tuned, but users did not need to be very versatile, and it was acceptable to use training to overcome some system shortcomings.

IT/MIS provided a service, crunching numbers and inventory records, and keeping track of historical data better than people could.

How are users' expectations changing?

TODAY'S USER EXPECTATIONS

User departments are drowning in the combination of budget restrictions, personnel shortages, labor shortages in key skills, and an increasingly demanding business environment.

PRODUCTIVITY, AND MORE PRODUCTIVITY

Because of cost pressures, there is a very significant emphasis on productivity in user departments. User personnel are being asked to operate efficiently in multiple applications. User departments expect their staff to be multifaceted, and fully knowledgeable in multiple, unrelated, applications like word processing, payroll, customer management, phone inquiries, and electronic mail.

Users read every day about decreased training time, enhanced context sensitive help facilities, intuitive user interfaces, and "push button" instantaneous context switches, and they are rapidly expecting IT/MIS to implement these more powerful environments into their everyday lives.

There is a dramatically increasing need for user proficiency, with a corresponding desire for reduced requirement for training. Modern training is expected to be painless, blindingly quick, and to be self-paced, without a required instructor.

The productivity of user personnel is now affected not just by training, or by experience, but by the complexity of the multiple application environments, and by the time it takes to switch contexts between them. One HP customer calculated that context switching alone - the time to switch from one application to another - was costing over $2M each year.

USER PARTICIPATION

In addition, more computer literate users are demanding a much stronger role in the design and development of their own applications. Many of these same users feel empowered to develop some applications on their own. This pattern places new training, documentation, and support requirements on the IT/MIS function.

Users no longer want single focus applications which require that they learn the quirks of the application. They want a user interface to be knowledge based, intuitive, and adjustable to match their behavior. They want their applications to blend naturally into all the work of the company.

MAKE IT, OR BUY IT?

Another subtle pressure is the "make / buy" decision. In the '60's, we "made" everything. We had to, because there was a limited 3rd party market. In the late 70's and early 80's, IT/MIS began suggesting the possibility of purchasing packages, and modifying them. This "buy" decision was often an opportunity to save a lot of money by eliminating the original invention cost. Unfortunately, it was nearly as often the cause of extensive budget over runs, when modifications cost more than an original implementation might have.

Times have changed with the development of a very viable 3rd party market. There are many more applications on the market, many of which are very directly applicable to common business problems.

Our users are rapidly embracing the perception that "buy" will work - "It says so right here in the paper, so it must be true". This puts IT/MIS in a very delicate position. If IT/MIS suggests a "make" decision, users ask why it can't be bought cheaper. If IT/MIS suggests a "buy" decision, users ask why we need such a big IT/MIS department, and "Is it installed yet and running exactly right, and why not?"

NEW DEFINITION OF QUALITY

In addition to all this, users are fully expecting unprecedented quality levels - much higher than in the past. "Quality" has been talked about, and written about, a great deal in the press, and in professional circles. Users now expect very few problems with new systems or applications and are very intolerant of down time or interruptions. As more users become technically "computer literate", this expectation will very likely increase. Budget and staff reductions in user organizations make outages much more expensive to the users, and any data loss is critical.

COLLABORATION

IT/MIS is increasingly expected to leave its comfortable role as a service provider, and enter a newer, more complex collaborative team player. This newer role requires much more than IT/MIS technology skills; organizational development, finance, business judgment, and customer management are increasingly important.

Let's look at how staffing issues are changing.

HISTORICAL EXPECTATIONS OF IT/MIS STAFF

IT/MIS staff have always been an unruly lot. Because they have had special knowledge they have demanded, and have usually received, some special treatment.

Historically, IT/MIS staff has expected a "good" job, with lots of variety, job enrichment opportunities, the ability to develop depth or breadth (personal preference), and growth potential.

Programmer analysts were competent if they had a good working understanding of COBOL, Screen management, and a data base. Specifications were usually technically developed, and given to programmers to implement.

The historical focus on staff management was retention of your best employees. In many cases, recruiting could be easy, depending on your environment and how much you were willing to pay.

What's changed here?

TODAY'S STAFF EXPECTATIONS

The major change has been the market's reaction to the decreasing numbers of qualified technical staff. Management focus, in addition to retention, must now be on recruiting requirements. In some ways, IT/MIS management must continuously recruit its current employees.

In the past, personal progress could have been measured by promotion, or by technical growth. Because of more vertically aligned organizations with flatter organization charts and fewer managers, the promotion path is not as available today. There are simply less managers, so there are less chances for promotion.

On the technical growth side, developers, programmers, programmer analysts now demand to work "at the state of the art". They ask for strong, nurturing environments, with superb tool sets, increased training, and skills development plans.

Tomorrow's technical staff will require much more than language, terminal, and data base knowledge. They will need sound business judgment, consultative skills, and a broad knowledge of all aspects of the business enterprise. With user departments capable of doing some of their own programming, the type of technical contribution made by IT/MIS staff will change a lot in the near future.

This feels quite similar to the needs of the past, but is different in one very important way. The decreasing numbers of qualified job applicants is making the job environment very competitive.

A different type of management style is increasingly required to attract, manage, and retain qualified technical talent. More focus on individual needs, job enrichment, personal growth, and the whole working environment is required today.

PRESSURE EVERYWHERE

Which brings us to the IT/MIS situation for the 90's. Pressure, pressure everywhere, and not much relief in site.

IT/MIS management must keep track of a dazzling array of technology changes, understand what they really mean to the company, measured in business terms, and become very effective at sharing that knowledge with the rest of the company. IT/MIS departments will need to keep up with technological change, *and* will need to learn to teach the value of that technology to their management partners, in ***business*** terms instead of technological advantages.

IT/MIS management is increasingly expected to play a stronger role in the corporate management team, making a direct contribution to the business. We must rise above the technology, develop strong management skills in business issues, build departments which have a strong positive effect on our company's bottom line, and learn to be a partner in our management team. Management of global issues will be continuously more important as this decade progresses.

IT/MIS management is increasingly also expected to collaborate with its users, no longer providing services, but pro-actively improving productivity in a rapidly changing, complex application environment. Improvements in the quality of IT/MIS deliverables will be a key success factor.

IT/MIS management must react to a growing shortage of technical staff, with the resulting need to really focus on staffing issues. Movement to modern development techniques, use of recent design and implementation tools, and use of relational databases will provide an exciting environment. Staff will demand more training, and new kinds of growth potential as organizations get flatter. Lots of creativity will be required here.

More communications skills are required today than have ever been required in the past. Negotiation skills, listening skills, and even the ability to teach will be very important.

More business skills, and business systems skills are required than have ever been necessary before. This will be especially true as IT/MIS managers are integrated into the mainline of the company management team.

In summary, IT/MIS must look forward, and develop new skills very quickly, to deliver on the expectations of upper management.

A shift from technology management to business management, and a new way of contributing to the management team, is increasingly important.

Tracking technological change is probably the largest current challenge. Technological change is happening so quickly that it is very difficult to track, and even more difficult to make "correct" decisions. Management wants decisions made on a "long term" basis, with predictable stable results. The technology world seems to be changing under our feet about every 9 months or so.

IT/MIS will need to do more training of its users, and will need to shift its role from service provider to technology mentor. Controls on user data will be minimized, and user access to data will be unprecedented.

And all this must be done in chronically tight fiscal situations, with an increasing shortage of qualified technical personnel.

The computer marketplace is incredibly volatile. Change is everywhere.

It is all happening at once, on many different axes, in many product areas, with global implications.

It is all happening with unprecedented speed.

And we are all in the middle of it!!! What an exciting time!!!

PAPER NO.: 2501

TITLE: Career Lifestyles in High Tech Jobs

AUTHOR: John Podkomorski
Hewlett-Packard Co.
c/o Ella Washington
19091 Pruneridge Avenue, M/S 46LG
Cupertino, CA 95014
408-447-1053

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

Paper Number: 2801
Employee Individual Development Plan

Bob Bliesner

Boeing Computer Services
P.O. Box 24346 M/S 6C-KE
Seattle, WA 98124
(206) 234-1418

Abstract

With labor costs taking a bigger bite of your computer dollar, it becomes even more important that our Data Processing Analysts maximize their productivity. In order to do that, an individual must be happy in their job, understand what the job requirement is, and be provided the necessary training to allow them to perform that job.

This paper will discuss how a Boeing Computer Services Organization of 13 analysts have used an "Individual Development Plan" (IDP) to assure that the analyst is working at a job they enjoy, understand the requirements of the job and receive the training required to perform and grow on the job. We will discuss the process used in detail, including evaluation of the process and how this process might be generalized for your environment.

Introduction

Just what is an "Individual Development Plan"? It is a written plan, prepared by the analyst and their management, which allows the employee and the job to grow into a more productive entity. It is not something which when done, is filed away never to be looked at again. To be meaningful it needs continuous updating to account for changes in the individuals goals, the organizations ever changing requirements and the changing business environments. As an individuals/organizations goals are achieved new goals must be established.

Why do we want to do this? Have we been doing this in the past? What has changed that makes it desirable to do this today? There have been many books written over the last few years addressing these questions. Peter Drucker has been discussing new management philosophies since the mod 50's which emphasis, among other things, the development of the individual. Tom Peters in his books and lectures indicates that the business world is changing, and we must allow our employees to be innovative in order to compete. They must take ownership of processes, and take pride in the work they accomplish. Jim Naisbitt in his books and lectures has indicated that there will be a shortage of personnel to hire in the future, so we must get better efficiency out of our existing personnel. Management theory, as taught in our education system, emphasizes delegation, which implies good communications, and that the employee knows what their job is and are trained to do it. We so often overlook the fact that dialog is required to establish good communication, and good communication is a requirement for job satisfaction. A lot of the things we will be discussing in this paper are not new, and maybe a number of these techniques you have been using for years. My intent is to expose you to a method that seems to work, by bringing employees and their managers together to strive for common goals. Hopefully this will result in happier employes being more productive in an environment when the goals of the organization are also satisfied.

Why Now?

When I started in the data processing industry (early 60's) machines were expensive and people were cheap and plentiful. Today a several thousand dollar PC has as much computer power as the big main frames of the 60's, and it is considerably cheaper that an analyst. This means that we must maximize the productive output or our analysts, and make sure that they are working on the right requirement. The other major item is our shrinking incoming work force. In the past there have been plenty of new graduates from our colleges and universities to satisfy our growing employee needs in the DP industry. The projections for the 1990's are not so rosy. The dip in the population of the US will hit the work force this decade, so it becomes even more important that we retain our existing work force, and make sure that they grow with the industry. The establishing of an Individual Development Plan and tracking that plan is one step in accomplishing that.

Boeing has recognized this and has a corporate thrust in employee development, to assure that there are sufficient resources in the future for the health of the company. They have stated as an objective that a collaborative individual development process between each manager and each of his or her employees will be established. We see many pilot projects being started with employee development emphasis to tackle this problem from many fronts. Specific formats of what this should look like have not been given, to allow each manager to do what works for their situation. As successes are recorded, such as the one I will be describing, we may see some more specific quidelines from the corporation.

My interest in this process came before the corporate directive. In raising our children, and in the extensive scouting work that I have done, the development of the individual has been of utmost importance. We have used extended camping trips (backpacking and residence camps) to teach kids self reliance, confidence and human interactions. When I became a manager, I had a strong desire to have all my employees reach their potential, and to stretch their capabilities. Dream where they would like to be, and begin on a path to make that dream come true. I had very supportive management, and the company objective just enhanced that position.

Getting Started

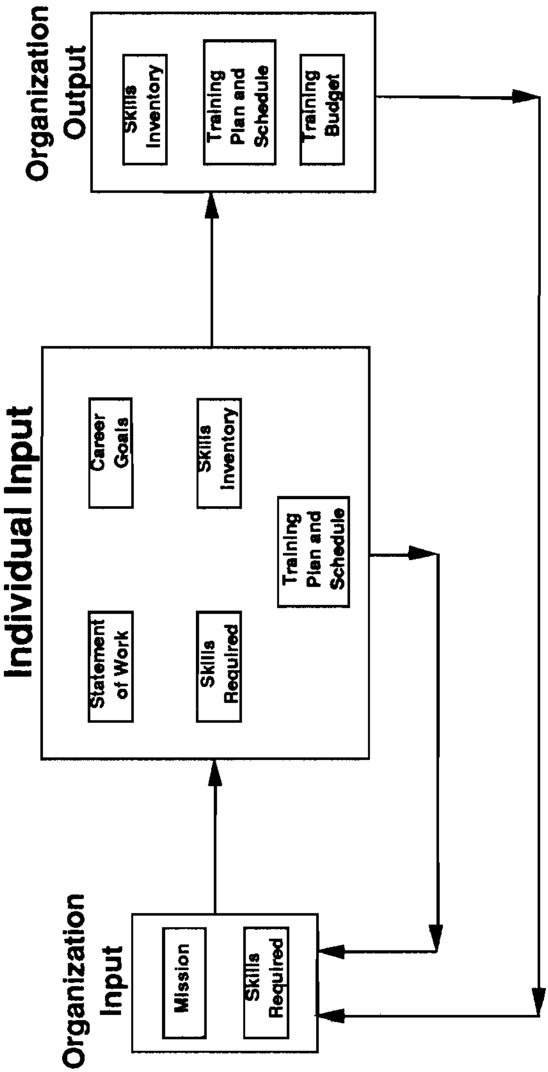
I got involved with this when I was asked to serve on a training plan team. Our manager, two levels above, was concerned that his organization did not have a formalized training plan, and training was taking place without clear direction. He formed a team, in the spring of 1989, of 5 managers and 3 non-managers to come up with some guidelines on putting together a training plan. The committee met weekly for roughly two months, and came up with a process flow of how to implement a formalized training plan. The intent was not to dictate form and function, but to give some guidelines on what to do. A copy of the process flow is found on the next page. Note that there are three major sections; the organizations input, the individuals input, and the organizations output.

For a training plan to be meaningful to the organization, the organization must have a clear understanding of what it is suppose to be doing (mission) and it must determine what skills are required to complete this mission.

The individual must determine what their role is in the mission, what are their plans for the future, what skills are required to satisfy both their role in the mission (statement of work) and their career objectives, and what skills they possess today. With that set of data it is then easy to determine what skills need to be developed and a plan can be put together on what training is required.

The third area is the organization output which consists of the skills needed to carry out the organization mission, and within the budget constraints, how much training the organization is willing to provide. This budget exercise must not include only dollars but labor time as well.

I liked the logic of this training process flow and used it as the basis for developing the IDP which is presented in this paper.



Employee Individual Development Plan
2801-4

Implementation

I worked very closely with my staff in implementing this IDP process. We built the process as we went along, not sure just how we would use the data collected. Had we known that it would be used only for the IDP process which we ended up with, the design would have been a little different. I had a team of 3 senior analysts which I used as a core group for detail discussions, and then we presented these items to the entire staff for review and modifications. Upper management was kept abreast of what we were doing, but were given no details. This team effort resulted in the 10 step process flow which follows:

1. Develop a mission statement for the organization.

2. Develop a skills matrix appropriate to the organization.

3. The employees write a statement of work for their specific job.

4. The employees write a Career Goals Statement, with Boeing emphasis.

5. The Employees fill out the matrix indicating the skill level their job requires, the skill level they have, and what skill level they desire to have.

6. The supervisor fills out the same matrix for each employee, except the desired column.

7. The supervisor identifies the differences between the employee skill assessment and the supervisors, and sits down with the employee and comes to an agreement on each of the levels.

8. The employee prioritizes the identified areas which need development, and comes up with any suggestions of how to solve the deficiency.

9. The supervisor works with the employee to form a formal IDP for the employee to follow.

10. The supervisor then uses the IDP from each of the employees to form a training plan within the budget constraints of the organization.

Now let's look at each of these 10 steps and give some details:

1. Mission Statement

The mission statement is intended to clarify why the organization exists, and must be tailored to your specific situation. An example of the mission statement we used follows:

Data Center and Spares Management Mission Statement 7/24/89

The mission of this organization is to provide information services of superior quality to our Data Center Customers and to our COSMOS* users while:

Providing services within budget constraints

Providing services within schedule

Teaming with the customers to provide the "Best" solution

Managing common user funding for baseline and internal systems

* Computerized Online System for Management of Supplies (Spares/Support)

2. Skills Matrix

The skills matrix is the key to the entire exercise. This determines the detail skills data you will be collecting and evaluating. We broke it into functional areas within the organization. We broke the skills rankings into only 3 skill levels of high, medium and low. There was also an implied 0 which means nonexistent skill level. We had considered using a five scale rating, and determined that for the first time at least, this would be more detail then we know what to do with. We requested the employees rate the skill level required for the job/task in there group. what their skill level is now, and what their desired skill level is for the future. Below is a sample of the type of item we included. The full matrix is available as a handout at the presentation.

FUNCTION			REQT LEVEL	EMP SKILL LEVEL	TASK DESIRED LEVEL
----------	---------	----------	------	------	---------
COSMOS-IC:					
	NEWPART 1&2,TRANS 1&2				
		Spiffy/Fortran			
		Transact/Fastran			
		View			
		Barcode			

3. Statement of Work

The statement of work as written by the employee is reviewed by the manager for accuracy and completeness, and if any differences surface they should be resolved at that time. This statement of work is very job specific. Below is a copy of the example we gave to all the employees.

Statement of Work

Jane Doe
8/8/89

Primary Duties and Responsibilities:

The employee will perform duties in the following areas:

Formulate statements of business problems and devise procedures for solutions of the problems through the use of electronic data processing systems. Confer with customers to define the data processing problem. Prepare charts, tables and diagrams to assist in analyzing problems, utilizing various business and mathematical techniques. Develop approach and technique to satisfy systems specifications. Conduct management presentation of findings. Analyze specific deficiencies in existing work methods and suggest redesign of work units and work flow to make better use of existing or proposed data processing systems. Devise procedures to solve problems through the use of electronic data processing considering capacity and limitations of equipment, operating time, and form of desired results. Analyze existing system logic difficulties and revise the logic and procedures involved as necessary. Develop logic and procedures to provide more efficient machine operations.

Provide consultation to other members of organization on both internal and customer applications.

Analyze data communications requirements, make recommendations, and assist in implementation of recommended solutions. Perform trouble shooting of data communications' problems.

4 . Goals

The goal statement should reflect both short and long term goals of the individual. The focus should be primarily on the job, but if the employee wishes to expand to his non-company life, that is their choice. The career goals statements are the employees, and not subject to management review. This has also been a topic of discussion as to just what the managers role should be in assisting the employee in their career goals. We have not reached a consensus on this issue. Below is a copy of the example we gave the employees.

Boeing Career Goals

Jane Doe
8/8/89

I would like to expand my knowledge of the HP3000, especially in the database and file structure areas. I would like to investigate management type of things, to make a clear decision if that is the way of my future. I am interested in learning more about program management and maybe being one in the future (if it appeals to me).

5, 6, and 7 The Matrix Exercise

The filling out of the skills matrix is done by the employee and the manager independently. The employee fills out the matrix form shown in step 2 above, while I took a short cut and filled out the summary matrix right on the PC. When the employee had their matrix completed they turned it into me. I then consolidated the input onto the summary matrix form, and had a discussion with the employee to go over the matrix in detail. This dialog then came up with an agreed rating for each of the skills in the matrix. I found this dialog one of the most useful activities in the process. It not only gave me a much better understanding of where the employee was headed, but also an excellent view of what the organization was or should be doing. Once the agreed skill level was established, both required, possessed, and desired, the arriving of a development plan, and a detailed organizations support training plan was a mechanical process. I would go through this skills list, and for any area where the skill possessed was less than the job required or the desired level, would place in the comment field a suggested training activity which would enhance this skill deficiency. Below is a sample of a section of this summary matrix. The full matrix is available as a handout at the presentation.

Jane Doe								
	Reqd			Skill Level			Des	Suggested Solution
FUNCTION	Supr	Empl	Final	Supr	Empl	Final		
COSMOS-IC:								
NEWPART 1&2,TRANS 1&2								
Spiffy/Fortran	3	2	3	2	2	2	3	OJT, seminars
Transact/Fastran	2	1	2	1	1	1	2	Adv. Transact
View	2	2	2	3	3	3	3	
Barcode	2	1	2	1	1	1	3	OJT

8. Developmental Areas

I then gave this matrix, of agreed values, back to the employee, and they then selected and prioritized the training they required/desired to meet their goals. This entire process at this point has been somewhat iterative, so if in the process they have clarified their goals or job statement of work, modifications were made reflecting the latest status. The list the employees prioritized included training requirements from all sources, not just company funded ones. I then took this list and build a plan which included all possible company supported training with specific class data and times. This list was normally larger than I had budget for, so my final training plan was this list reduced, based on budget resources both dollars and labor and on the employees prioritization of training requirements. Below is a sample of the full employee prioritized list of training items which are requesting company resources. Items not requiring company resources were not included in this list.

1990 Training Plan Master List (Full employee Request)

Class/Seminar	Location	Dte Cls Pln	Sch	# Das	Org	Emp Lame	O/D	Regis. $ Planned	Total Hours Plan	Act	Travel Cost Plan	Act	Type
Info Engin. Smnr	??	Jun-90		5	4832	Emp 1	O	1500	40		1,360		CON
IDEF Workshop	B0468	Feb-90		5	4832	Emp 1	O	825	40				E&T
MFG		Aug-90		2	4832	Emp 1	O	0	16				INT
Team Building	BCS356	Apr-90		2	4832	Emp 1	O	330	16				E&T
MGT Interex	Las Vegas	Mar-90	Mar-90	3	4832	Emp 2	O	375	24		1,030		CON
Teaming	BCS356	Apr-90		2	4832	Emp 2	O	330	16				INT
Listening	BCS722			1	4832	Emp 2	O	210	8				CON
SWEIG	Bellevue	Jan-90	Jan-90	0.3	4832	Emp 2	O	0	3		0		INT
Uncom. Lead	B0262	Feb-90	Feb-90	0.3	4832	Emp 2	O	0	3		0		INT
Max. Mgt Prm	B0263	Mar-90	Mar-90	0.3	4832	Emp 2	O	0	3		0		INT
TBD	Video			0.3	4832	Emp 2	O	0	3		0		INT
TBD	Video			0.3	4832	Emp 2	O	0	3		0		INT
TBD	Video			0.3	4832	Emp 2	O	0	3		0		INT
Leadership	Las Vegas	Mar-90		2.5	4832	Emp 3	O	0	20		0		INT
MGT Interex	Las Vegas	Mar-90		3	4832	Emp 3	O	450	24		1,030		CON
Mgt. Glob. Chg	B0261	Jan-90	Jan-90	0.3	4832	Emp 3	O	0	2.4		0		E&T
Uncom Lead	B0262	Feb-90	Feb-90	0.3	4832	Emp 3	O	0	2.4		0		E&T
Max Mgt Prm	B0263	Mar-90	Mar-90	0.3	4832	Emp 3	O	0	2.4		0		E&T
Oracle Seminar	Bellevue	Feb-90		0.5	4832	Emp 3	O	0	4				VEN
Oracle Intro	B0024	Jun-90		5	4832	Emp 3	O	825	40				E&T
				33.7				4845	273.2		3,420		

The actual training plan would then be the portion of the total list which could be supported by the organization.

9. Formal IDP

Once the employee had prioritized the required/desired training, they were ready to formalize the IDP. They did this by listing the skills and deficiencies they wanted to work on, and if they accepted the matrix in its entirety, some elected to just reference the matrix. The actual IDP contains all training opportunities, not just the ones requiring company resources. Below is a copy of a blank IDP form which we used.

Individual Development Plan (page 1)

Employee Name:	Date:
Organization Mission:	
Employee's Statement of Work:	
Employee's Career Goals:	
Skills Required:	Skills Possesed:

Individual Development Plan (page 2)

Employee Name:	Date:
Skill Gaps:	
Solutions (include what, where, how and the provider):	
Signatures:	
Employee Date	Supervisor Date

10. Training Plan

A copy of the final training plan for the organization for 1990 is included in the appendix. I made it clear that the training plan was the organizations and we would work it, but the Individual Development plan was the employees, and they should be working that. I felt so strongly about this that I signed their IDP but did not keep a copy of it.

Evaluation

The collecting and analyses of this data is not free. We budgeted 10 hours per employee, and used on the average of 6 hours. It took me 4-6 hours per employee to do the details, as well as roughly 40 hours on the front end setting things up. Because all our analysts are busy people we had to force the exercise by some rigid time lines. I had a management commitment to complete the entire organization in 1989, so I made up a chart which had two interviews a week, and had employees sign up for the week they wanted to review the data. This worked out very well.

Overall, the experience was very well received. It provided me a much better understanding of the organization as well as an indepth knowledge of the individuals future plans. It was established that several people didn't have a desired future in the organization, and I helped those individuals find another position in the company which satisfied their short and long range goals. Several individuals felt that their short term goals could be satisfied by the organization, but their long term goals were outside the organization.

No attempt was made to use any of the data for performance evaluations, although it would be very easy to do that. It was not the intent of this IDP to be used for performance evaluations.

The communication between the manager and employees was greatly improved through these dialog discussions. I spent an average of 2 hours in dialog with each employee on this subject. I also found that when the annual work performance review was done, we talked a lot about their IDP's.

As a last step of the IDP process, I discussed with each employee how the process we had just completed could be improved. The suggested improvements came in the three major categories:

- Communications
- Format
- Focus

Communications

Some of the terms used in the matrix, were either new to the employees, or the meaning wasn't clear. This needs to be resolved by either, increasing employee involvement on the front end definition of the matrix, or include a glossary of terms, or maybe both.

Format

This was the most common improvement stated. A section on language skills needs to be included, which is independent of the applications. Each of the application areas should have the language items removed, and include the following:

Data Base Knowledge
Dictionary Knowledge
Knowledge of Internal Code of Programs
Knowledge of View Screens
Understanding Customers Job
Knowledge of the Application

The soft skills need to be broken out in a separate section, and expanded. This includes such skills as presentations, listening, leadership, etc.

The section on possible future requirements, should be changed to possible future focuses, and consider removing the supervisor column of this section.

In constructing our matrix, we left the skills for one specific job off completely, thereby making the process for the employee involved sonewhat meaningless. Be careful not to do that.

Focus

When we started the matrix, we knew we wanted to collect a lot of data, but were not clear on how we were going to use the data. At the completion, some of the confusion was caused by this lack of focus. The matrix should be revised to focus on training and skills levels on an individual basis. If other types of data are required, then use another exercise to collect that data.

Lessons Learned

This exercise was an excellent communication tool between the manager and employee. The dialog was extremely effective. The follow through will be important. If neither the manager nor the employees follow through with periodic reviews, the exercise will loose some of its meaning. Also, as employees move into and out of the organization, a process must be in place to either redo this exercise, or to carry the data to the new organization.

Some difficult realizations can come from this exercise. We had the case of an employee performing well on the job and enjoying the job, but they were not working in their desired area (we are a maintenance shop, they wanted a development shop using CASE tools). As a direct result of this exercise we lost a good employee from the organization, however did retain them in the company. I think that the company won and the employee won in this situation, but the organization lost. In another case it became obvious to the employee that they would be happier in another position. In this specific case they were a misfit in our organization and were struggling from a performance standpoint. We found another position within the company and in this case all people and organizations won. I also found it very satisfying to be able to help people grow in their careers, in their desired fashion, while understanding and enhancing the organizations goals.

Conclusions

Other managers have done similar things in implementing their IDP's. A number did not use the skills matrix, due to the time involved in implementing the process. At least one who did not use the matrix, plans to use the matrix next time. If I were to repeat the IDP process from scratch, I would definitely use this technique again. Our general feeling is that the matrix exercise would be meaningful if done on a annual basis, with formal IDP reviews being done quarterly. Informal reviews of the IDP should be done monthly, or whenever a change, which affects the IDP parameters, takes place (organization change, change of work assignment, completion of goal, change of management, etc.).

Remember that these forms and the techniques described here are a way to accomplish a formal IDP between the Employee and their manager. The important thing is that this understanding takes place, and that the individual has a plan for their career, not that a set number of forms are filled out. Have fun adapting these ideas to your situation and environment, and good luck on your IDP process.

Appendix:

Training Plan for Org (Page 1)

1990 Training Plan Master List

Class/Seminar	Location	Date of Class Pln	Date of Class Sch/Act	# Days	Org	Employee's Last Name	O/D	Regis. $ Planned	Total Hours Plan	Total Hours Act	Travel Cost Plan	Travel Cost Act	Type	Reg In	Status
Info Engin Sem.	??	Jun-90		5	4832	Emp 1	O	1500	40		1,380		CON		Pln
MFG	Bellevue	Aug-90		2	4832	Emp 1	O	0	16		0		E&T		Pln
MGT Interex	Las Vegas	Mar-90	Mar-90	3	4832	Emp 2	O	125	24		1,030		CON	Jan-90	Sch
Team Building	BCS358	Apr-90		2	4832	Emp 2	O	330	16		0		E&T		Pln
SWEIG	Southcenter	Jan-90	Jan-90	0.3	4832	Emp 2	O	0	3		0		CON	Dec-89	Done
Uncom. Lead	B0262	Feb-90	Feb-90	0.3	4832	Emp 2	O	0	3		0		E&T	Dec-89	Sch
Max. Mgt Prm	B0263	Mar-90	Mar-90	0.3	4832	Emp 2	O	0	3		0		E&T	Dec-89	Sch
TBD	Video	May-90		0.3	4832	Emp 2	O	0	3		0		E&T		Pln
TBD	Video	Jun-90		0.3	4832	Emp 2	O	0	3		0		E&T		Pln
TBD	Video	Jul-90		0.3	4832	Emp 2	O	0	3		0		E&T		Pln
MGT Interex	Las Vegas	Mar-90	Mar-90	3	4832	Emp 3	O	425	24		1,030		CON	Jan-90	Sch
Mgt. Glob. Chg	B0261	Jan-90	Jan-90	0.3	4832	Emp 3	O	0	2.4		0		E&T	Dec-89	Done
Uncom Lead	B0262	Feb-90	Feb-90	0.3	4832	Emp 3	O	0	2.4		0		E&T	Dec-89	Sch
Max Mgt Prm	B0263	Mar-90	Mar-90	0.3	4832	Emp 3	O	0	2.4		0		E&T	Dec-89	Sch
Leadership		Ongoing	Ongoing	2.5	4832	Emp 3	O	0	20		0		INT		Done
Oracle Seminar		Sep-90		0.5	4832	Emp 3	O	0	4		0		VEN		Pln
VPLUS/3000	Seattle	Oct-90		4	4832	Emp 4	O	900	32		0		VEN		Pln
IV League	Anaheim	Apr-90	Apr-90	4	4832	Emp 4	O	660	32		1,195		CON	Jan-90	Sch
Data Comm	B0047	Aug-90		3	4832	Emp 5	O	495	24		0		E&T		Pln
Data Communic	B0046	Apr-90		2	4832	Emp 5	O	330	16				E&T		Pln
Peop to Peop	BCS792	May-90		1	4832	Emp 5	O	165	8				E&T		Pln
Sys Mgt Skls	Seattle	Apr-90		5	4832	Emp 6	O	1300	40				VEN		Pln
Team Building	BCS356	Mar-90		2	4832	Emp 6	O	330	16		0		E&T		Pln
Time Mgt	BCS662	Sep-90		1	4832	Emp 6	O	165	8				E&T		Pln
Listening	BCS722	Apr-90		1	4832	Emp 7	O	165	8		0		INT		Pln
Sigrapid	??	Mar-90		3	4832	Emp 7	O	300	24		1,030		CON		Pln
Word	B0150	Jun-90		2	4832	Emp 7	O	330	16				E&T		Pln
Prog Mgt	Seattle			3	4832	Emp 7	O	0	24				E&T		Pln
Case Semin	??	Apr-90		3	4832	Emp 8	O	1000	24		1,030		CON		Pln
AI (BCS314)	Seattle	Mar-90		5	4832	Emp 8	O	825	40		0		E&T	Jan-90	Sch

Training Plan for Org (Page 2)

1990 Training Plan Master List

Class/Seminar	Location	Date of Class Pln	Sch/Act	# Days	Org	Employee's Last Name	O/D	Regis $ Planned	Total Hours Plan	Act	Travel Cost Plan	Act	Type	Reg In	Status
Nowrug	Seattle			2	4832	TBD	O	50	16		0		CON		
Image				5	4832	TBD	O	0	40				INT		
DXC				1	4832	TBD	O	0	8		0		INT		
Oracle	B0024	Mar-90		4	4832	Emp 9	O	660	32		0		E&T	Jan-90	Sch
Intro to C	BCS397	Oct-90		5	4832	Emp 9	O	825	40				E&T		Pln
MPG	Bellevue	Jun-90		2	4832	Emp 10	O	0	16		0		INT		Pln
Network Conf	??			5	4832	Emp 10	O	1000	40		1,360		CON		Pln
NOWRUG		Feb-90		2	4832	Emp 11	O	50	16		0		CON	Jan-90	Sch
Oracle Seminar		Feb-90		0 5	4832	Emp 11	O	0	4				VEN		Pln
Oracle RDBMS	B0024	Feb-90		4	4832	Emp 11	O	660	32				E&T	Jan-90	Pln
PAC IV	Seattle	Jun-90		1	4832	Emp 11	O	50	8		0		CON		Sch
Mark IV Inter	BCS718	Apr-90		5	4832	Emp 12	O	825	40				E&T		Pln
JCL Inter	BCS063	Oct-90		2	4832	Emp 12	O	330	16		0		E&T		Pln
DXC				1	4832	TBD	O	0	8				INT		
Image				5	4832	TBD	O	0	40				INT		
TBD				2	4832	TBD	O	0	16				INT		
Leadership	BCS113	Apr-90		2	4832	Emp 13	O	330	16		0		INT		Pln
Interex	Boston	Aug-90		4	4832	Emp 13	D	0	0		0		CON		Pln
Team Building	BCS356	Sep-90		2	4832	Emp 13	O	330	16		0		E&T		Pln
SWEIG	Southcenter	Jan-90	Jan-90	0.3	4832	Emp 13	O	0	2.4		0		E&T	Dec-89	Done
Uncom. Lead	B0262	Feb-90	Feb-90	0 3	4832	Emp 13	O	0	2.4		0		E&T	Dec-89	Sch
				114.8				14455	890	0	8,035	0			
	Cost Model			118.5				13301	948		8,250				

Paper No. 2802
THE SEVEN DEADLY SINS IN HIRING EXECUTIVES, MANAGERS & PROFESSIONALS

Richard J. Pinsker
PINSKER and COMPANY, Inc.
P.O.Box 3269
Saratoga, CA 95070
(408) 867-5161

Every one of us can point to at least one subordinate whom we have hired who should never have been hired in the first place. Some executives have told me that as many as one-third of their work force might fall into this category. Whatever the numbers, it is safe to say that when it comes to the most important resource in a corporation, the human resource, we fail to make the best selection decisions. And this should come as no great surprise.

When employees are promoted to managerial positions, they are ordained with certain managerial powers. One of these is the power to make hiring decisions. Unfortunately, this power comes sans the knowledge or training on how to do it. At this point in one's career it becomes embarrassing to ask, "How do I hire someone?"

The often severe shortage of certain professional categories of personnel compounds the problem. On the one hand, there are not enough candidates to fill all the vacancies. While on the other hand, you need to attract, select and hire the best people for your organization. Can you make a sound selection decision if you do not have candidates from which to select? Of course not! However, by developing and implementing a proactive recruitment and selection program, you can improve the quality and the quantity of people you hire. Additionally, an aggressive program will put you steps ahead of the competition for these candidates.

A proactive recruitment and selection program demands two vital elements:

(1) A campaign and procedure for the continual identification of candidates and sources for candidates; and

(2) A management staff, well trained in the skills and techniques to make sound, objective hiring-decisions.

The first starts with a company-wide philosophy and goal, while the second requires training in and the implementation of a disciplined selection process.

Through my executive selection consulting practice, I have uncovered the "mistakes" executives most commonly make when recruiting key employees. These mistakes are not just confined to inadequate interviewing techniques. Rather, they represent the total recruitment process. Let's take a look at each of the "Seven Deadly Sins" and how a logical selection process, the P.I.E. Selection System, can overcome them.

1] ***Not properly defining what is needed.*** This is probably the most deadly of the sins. The tendency is not to be specific in defining the requirements. (Perhaps that will allow most anyone to be accepted or rejected.) "I just want a good MIS Director" said one executive. (She wouldn't know a "good" one if she saw one.)

The document used most frequently to define a job requirement is the job description. At best, it defines duties and responsibilities; activities to indulge in rather than accomplishments to achieve. "Will be responsible for developing and implementing a fully integrated management information system" may be an oversimplified example. What systems would we want to include, and in what priority would we want them defined and implemented?

Perhaps more indicative of the job requirement would be a measurable statement, a performance expectation, such as, "Will provide management with an integrated customer quotation and order processing system within six months". Sounds familiar? Of course; just another application of management by objectives. By defining a hiring requirement in this manner, we have created more objective criteria to measure a candidate's experience and potential. Looking at candidates who have "developed order processing systems" would selectively narrow the list of candidates from those who have just "developed management information systems."

Every position can be defined by performance expectations. This includes engineers, scientists, and manufacturing, financial, administrative, sales and marketing professionals. No exceptions! If you cannot identify the specific results you expect a person to accomplish within a given period of time, then you really need to question your need for hiring someone.

Defining performance expectations is the beginning of the P.I.E. Selection System. The system has three equally important phases:

Profile - A results-oriented definition of the candidate requirement, to be used as a guide for the next two phases;

Interview - The means to uncover, in-depth, the skills, capabilities and personality of the candidate relative to the requirements defined in the Profile;

Evaluate - A confirmation process to ensure the findings and hypothesis developed during the Interview.

The P.I.E. Selection System is a closed-loop process that leaves little to chance in making a hiring-decision.

A list of eight to twelve of these performance expectations, results to be achieved during the first twelve months, serves as a guide when interviewing and evaluating candidates. This guide is the Profile; the standard against which we measure all candidates. Do the candidate's skills, accomplishments, and personality suggest that he or she could meet these performance expectations? This question should be answered during the Interview and Evaluate (reference checking) phases. Therefore, all people who interview the candidate should have a copy of the Profile so that each knows exactly what to look for during the interview. This Profile also provides a common, objective measure for evaluation and selection.

2] Unintentionally limiting the sources for candidates. Like most good executive search consultants would do, a rifle-shot approach is much more efficient and successful than a buck-shot discharge. And the best rifle-shot target, and one of the most effective and inexpensive sources for candidates, comes as a result of networking. If you have 50 employees, you have 50 recruiters on your payroll. Each employee should be referring candidates, as well as sources for candidates, to a central data bank of names. These become the target for your rifle.

Many organizations pay bounties to their employees for each referred person who is hired. If the employee does not get many "hits" the reward system may fizzle. A more proactive approach is to build the recruitment function into each employee's performance goals and performance reviews. This would emphasize that recruitment is part of everyone's goals and objectives.

And the exclamation point to that statement lies in their bottom line; salary increases and bonuses based on meeting performance goals.

Professionals and managers have significant opportunities for networking. Develop a networking attitude among your employees so they always will be on the lookout for prospective candidates and sources for candidates. Conventions, trade shows, seminars, association meetings, colleagues, etc. are just a few of the places for finding names. For example, when technical people attend conferences, seminars or meetings, they should note the attenders and presenters as well as the program content itself. Authors of articles in your field are also good contacts who could be helpful in pinpointing candidates. Then provide a system for collecting and retrieving this information, and a viable recruitment tool is at your disposal.

3] ***Failing to interview candidates thoroughly.*** A quick trip to any bookstore will tell you why candidates are better prepared for the employment interview than most managers. Scores of books enlighten the candidate on how to answer to your questions. Resumes are prepared by professionals. Outplacement firms teach the candidate what to expect and how to respond. Image consultants tell the candidate what to wear and how to act. All this preparation culminates in a face to face meeting with the often unprepared, hiring manager.

Hiring decision-makers must develop the skills necessary to conduct a thorough interview. Basic to the interview is a sound understanding of the requirements of the position - the Profile. Overlay on those requirements an interview agenda; one which traces the candidate from education to the present time. Do not ask questions which may be answered by a yes or no. You learn very little about the candidate. Instead, ask questions that stimulate discussion, (what, how, why), and make use of active listening gestures that will keep the candidate talking about 80% of the time. Taking notes is a must. It shows interest and helps in the evaluation.

One excellent interviewing technique is to pose a real problem for the candidate to solve. The way the candidate approaches the problem, the questions asked, and the logic of the thought process are more important than the actual solution.

An in-depth interview should provide a history of the candidate's accomplishments, a feeling for compatibility of the candidate's style and "chemistry", and an understanding of the way the candidate thinks. Based on the interview, assumptions can be made that will indicate whether the candidate can meet the defined performance expectations as stated in the Profile.

4] ***Falling for the "Halo Effect."*** We have been inundated with examples of the "Halo Effect" since early childhood. Product endorsements by celebrities are good examples. The picture of a well known model on a cosmetics package implies that the model: a) knows what is best for your skin; and b) you will be more attractive (like the model) if you use that product. The "Halo Effect" simply means that because a person is outstanding in one area, they will be equally outstanding or are very knowledgeable in another. However, that can be a devastating assumption when making a hiring decision.

The systems analyst who does an outstanding job may not be the best person to be promoted to a manager. We should not assume that stellar technical performance automatically bestows managerial skills. Likewise, the MIS executive of a major division or corporation may not be the best candidate to head up an emerging department in a small start-up group or company. Not everyone has the entrepreneurial skills and hands-on orientation to participate from ground zero.

5] ***Wishful thinking.*** This sin usually comes about when we desperately need to fill a position. Every shortcut in the book is employed. The logical selection steps are totally ignored. The "Halo Effect" often prevails in the decision. Hasty promotions are the norm.

The wishful thinking excuse can be easily recognized in these favorite statements. "I don't have the time to go through all this now". "Nancy deserves the promotion and could take over the job on Monday, (even though she doesn't have all the skills)". "We need somebody now!".

This sin can be avoided rather easily. Plan ahead! Begin now by creating a data bank of sources and candidates. Develop and practice a selection system that insures nothing falls between the cracks. Then make sure everyone sticks to and follows the system. Combine a system check list with the hiring requisition approval form. Do not approve a new hire until you are convinced that all the steps on the check list were followed.

Considering the tremendous cost potential of a poor hiring decision - lost opportunities, decay in morale, wrongful discharge litigation, increased turnover, - hiring shortcuts just don't make sense.

6] *Ignoring Intuition.* As much as we would like the selection process to be logical and objective, intuitive feelings should not be ignored. Intuition is the perception of truth without the benefit of reasoning; at least on the face of it. It has little to do with logic. Or does it?

Intuition is often based upon fact and experience on a subconscious level. Perhaps a candidate reminds you of someone you have worked with in the past. Or some mannerism or response by the candidate triggers a feeling that does not jibe with what the candidate said.

Intuition has always been a powerful force in business. Most successful executives use it in their decision making process. Go through the logical selection steps in making a hiring decision. Then, if something still "bugs" you, follow your intuition. "When in doubt, don't!"

7] ***Failing to check one more reference.*** Some employers only check a few references, while others check none. "Checking references is a waste of time. No one will say anything bad about a former employee. They're afraid of a lawsuit."

Absurd advice! Reference checking is as important as every other step in a good selection system. But to be valuable, there are a few rules. If followed, you will have the knowledge to verify any hypothesis you have developed on the candidate's capabilities and compatibility with your firm.

First Rule - <u>Always ask each person who serves as a reference for one more name.</u> You typically start with the references given by the candidate. Then expand on this list until you are certain that what you see is what you are really going to get. This should include discussions with the candidate's peers, subordinates and superiors. This should not include discussions with the human resources department, unless of course, the candidate is a human resource person.

Last Rule - <u>Handle each reference discussion just like an interview.</u> Ask the type of questions you would ask during an interview. Press for specific accomplishments that the candidate achieved.

Here are the ten most important questions you <u>must</u> ask each employment reference:

1. How and when did you know the candidate, and what was

your relationship to the candidate?

2. Trace the candidate's progress/relationship with you/your company.

3. What were the candidate's most significant accomplishments? (You want individual, specific results, not departmental accomplishments.)

4. What were the major strengths you noted in the candidate?

5. We all have some shortcomings or areas that can be improved upon. What were the candidate's shortcomings, and how did she/he accommodate them?

6. Why did the candidate leave your company? (Or, do you know why she/he might be thinking of leaving?)

7. How would you describe the candidate's relationships with others? (This includes peers, subordinates, customers, and any others who might be appropriate.)

8. We are considering the candidate for a position that involves (describe performance expectations for the position). From your experience working with the candidate, how do you feel she/he would contribute to and meet these performance expectations, and why do you feel that way?

9. In summary, how would you describe the candidate's employment with your company and where would you rank the candidate in comparison to others in similar positions? Would you want to work for or with this person again? In what kind of environment?

10. Who else in your company should we talk to about the candidate?

These reference interviews are as important as the personal interview with the candidate. When possible, meet face-to-face with the person providing the reference.

One more significant point about reference checking. Just because a candidate is known to you personally, or has been referred by someone you trust and respect, should not excuse you from doing a very diligent job of checking references. Do not let the "Halo Effect" cloud your better judgement!

Each of the above deadly sins, if ignored, represents a significant flaw in a selection system. Creating a Profile by defining candidate requirements in terms of results rather than tasks, and then diligently using this Profile during the Interview and Evaluation phases, should significantly improve your hiring-decision making skills, and thus your selection results.

The '90s will continue to be a very difficult decade to find good people. With companies operating in a more "lean and mean" mode, **every** hire takes on greater significance. Taking a proactive approach to recruitment and selection by making everyone of your employees "recruiters" for your company; and adopting a logical, objective total selection system with management well trained and required to use it, will provide you with a decided advantage over other companies.

SYSTEM WISDOM

WHAT YOU WON'T FIND IN COMPUTER MANUALS!

Robert Mattson
9545 Delphi Road S.W.
Olympia, WA 98502
(206) 352-5038

Introduction

Oliver's Law: "Experience is something you don't get until after you need it."

"Programming is part art, part science and certainly part philosophy." Zack Urlocker

There is an interesting phenomenon in systems and computer science. We tacitly hold to the belief that there are few system laws or rules. The few we do acknowledge we seem to believe are best learned through experience's "hard knocks." Not surprisingly, we don't spend much time discussing, documenting or teaching any system rules.

The most telling evidence comes from recent computer science graduates. If you ask such a person to tell you the fundamental axioms and rules of systems they will stare at you blankly. They won't know what you're talking about. They were never taught any system rules.

Is this the way it should be? Are there no rules that underlie the workings of all systems? What about good old familiar Murphy's Law. We laughingly nod in agreement when some disaster happens which this rule seems to foretell.

As Yogi Berra says.. "You can observe a lot by just watching." From my watching and involvement over a fair number of years, I've observed some important basic rules that apply to systems.

The significance of these rules is as follows.

1) These rules are very valuable in the designing, building and maintaining of systems. They provide important guidelines about how things should be done.

2) Many of the problems we have in systems today are attributable to the lack of understanding and/or application of these rules. This is most obvious with the less experienced. What is even more surprising are the number of "experienced" people who seem to be oblivious to these rules.

3) There are great benefits in finding, exploring, and documenting these rules. These benefits are maximized when we start teaching these rules to new system professionals.

My goal here is to present a proposed set of the system rules. The hope is that this spurs others to expand and clarify the set. Hopefully these rules will then be taught to and applied by neophyte system professionals. As a consequence, they will miss "re-inventing" many classic mistakes and improve the quality of what they produce. This last point is the telling one. If we can only learn by hard knocks then we will never progress beyond the stone-age in systems. All "sciences" have progressed because people discovered, recorded and taught the rules that underlie the discipline. This allowed later people in the field to start further along the road of knowledge. Surely, we can't afford for each new system's person to start from point zero.

The Nature of Systems and Rules

"Everything can be viewed as a system."

I use the word "systems" to describe all aspects of computer systems and their associated environments. My definition is meant to be broad and inclusive. It includes the people designing, building, maintaining and using the system. It also includes the tools, hardware, management, etc.

What is a rule? The dictionaries defines a rule as... "A statement that describes what is true in most or all cases." This is an excellent definition.

Now, a few of you may want to take issue with a rule and point to an instance of when it didn't apply. Your experience could have been such. Remember, I'm talking about tendencies and probabilities. For each rule presented there are cases where it doesn't apply or another rule has precedence. Yet in my experience 90 percent of the time the rules do apply. When I find myself dissatisfied with how things are going or the results; I can usually point to my having gone against one of these rules. That's why I believe these rules are "true."

For ease of understanding in this paper I've grouped these rules within areas such as "general", "people", etc.

GENERAL RULES

Here are some rules that are so basic that they apply to all the other rules. Maybe these are the fundamental axioms of systems.

Inter-relate Rule : Everything Is Inter-related.

Everything is inter-twingled." - Yourdon Training Manual

"When you try to pull out anything you find it's tied to everything else in the universe." John Muir

This rule says that any piece of a system is connected to every other piece. This may seem fairly obvious when you think about the computer code in a single program. It may not be as obvious when relating the personnel policy on payment for overtime to the same code. In the latter seemingly unrelated case, for example, the two factors can have a very significant direct cause and effect relationship.

So, some connections are obvious, strong and highly inter-dependent. Others are subtle and weak. The key is that no two pieces of a "system" are independent.

I see a lack of understanding of this rule in inexperienced people. They tend to see each piece of a system or system project or personnel issue as standing alone.

Applying the rule has many benefits. The most common is to ask "What other things will affect this?" or "What other things will this affect?" As with most of the rules, just asking the questions thoughtfully can give surprising answers and benefits.

Change Rule : Change Is For Sure.

"The one constant is change."

This rule says things in and about a system will change. Understanding this rule is extremely important.

Yet countless times I've seen system designs and products which would seem blind to this rule. This results many times in wasting large amounts of work and dollars on future enhancements, fixes, etc. For example, how many of you have spent long hours changing hard-coded constants or array limitations or acceptable type sets?

One of the most common manifestations of applying this rule is that systems need to have flexibility and adaptability designed and built into them. We know, from this rule, that no matter how perfect a design is today that things will change tomorrow. We seldom know exactly how it will change. Yet, like our comment about the Inter-related Rule, just asking the question "How can I provide for flexibility and adaptability?" can result in amazing benefits.

Two Sides Rule : There Are Two Sides To Every Issue.

"There ain't no such thing as a free lunch."

George's Law "All pluses have their minuses."

How often do I see people being sold or selling something in systems when the only side that is being discussed is the "plus" side. Decisions can be made if the other side is discussed. Most times they will be better decisions.

This is not the same as nay saying or negativism. It is understanding that we need to look at many sides to a decision or problem in order to improve the resulting quality.

Surprisingly, many times the "discussion" really can just go on in a single person's head. Just by asking the question "what is the other side of this issue" can result in better decisions and results.

Reasons Rule : Justifications Are Usually A Comfort Issue.

"The hardest place to leave is our comfort zone."

Whenever I hear someone justify why some decision should be made one way or another, I always look for the comfort zone. This means most people most of the time will suggest solutions that are based a lot on what they feel most comfortable about.

I suggest that the current "popularity" of the "C" language is in large part due to the huge number of recent graduates who feel comfortable with it. Same with the UNIX operating system or COBOL. Nevertheless, I've heard and read elaborate explanations to justify what is just a comfort zone issue.

The key here is that just because someone is comfortable is not necessarily the best reason to make some decision. Without recognizing this clear bias, too many system decisions are made on "facts" that aren't justifiable.

Assumption Rule : Assuming Is Dangerous.

"Assumption is the mother of all foul-ups."

"Assume and you make an "ass" out of "u" and "me."

I wish I had a dollar for every time I heard someone say "I assumed..." If one could magically observe the events preceding this statement they would most likely be described as some type of "mess-up" having occurred. So if it's such a common phenomenon why do we assume so often? Especially when many assumptions are so easy to verify.

Every system person should have a little bell go off when he/she consciously or subconsciously assumes. This would make them pause for a moment and make a better judgement as to whether they should verify that assumption.

PROJECT RULES

Team Size Rule : The Best Systems Come From Small Teams

Brook's Law: "Adding personnel to a late software project makes it later."

As you increase the project team size beyond five the productivity and quality goes down dramatically. Stated in a slightly different way; throwing people or money at a project in trouble will probably make it worse.

This rule has been "known" for almost thirty years. Yet so often these days I read or hear of some system disaster that included a team of hundreds.

We ought to be thinking; if I can't do this with five people something else is wrong.

Body Rule : One Good Person Is Worth A Ton Of Bodies.

"No matter what they tell you it's always a people problem." Gerald Weinberg

A programmer is not a programmer is not a programmer. With all do regards to Shakespeare, one programmer or analyst-designer is not the same as any other. The range in system professional capabilities is at least ten to one. We should always go for quality over quantity in system project staffing.

Time Rule : Tasks Take Longer Than Planned.

Warren's Rule: "To spot the expert, pick the one who predicts the job will take the longest and cost the most."

Van Gogh's Law: "Whatever plan one makes, there is a hidden difficulty somewhere."

Murphy's First Corollary: "Nothing is as easy as it looks."

Golub's First Law of Computerdom: "Fuzzy project objectives are used to avoid the embarrassment of estimating the corresponding costs."

Golub's Second Law of Computerdom: "A carelessly planned project takes three times longer to complete than expected; a carefully planned project only takes twice as long."

A rookie mistake is believing tasks can be done in too little time. Surprisingly, this is also a common veteran's mistake. This is very confusing. I understand why rookies make the mistake but why veterans?

One hypothesis has to do with the fact that few people keep honest enough track of their time to know exactly how long it takes them to do anything. Therefore they have little chance of learning from past experience. This will only be cured when we start tracking what we do and how long we do it. And it will have to be done in a way that will allow us to use this data later.

Another factor is that no two system projects are ever exactly alike. They have different people, time of year, experience level, tools, etc. Therefore it is unreasonable to expect the projects to take the same amount of time, cost the same or produce the same quality result.

If you know this rule you can save yourself a great deal of embarrassment or worse. This is true whether you're the project leader, manager or the person doing the work.

ANALYSIS-DESIGN RULES

Know Rule : Know That You Can't Know Everything.

Wooden's Law: "It's what you learn after you know it all that counts."

"If you can't tell me three things that you don't understand about the problem... you don't understand the problem."

"Seven-eighths of everything is hidden."

You can never know and understand everything about a system problem, technique or design; so you need to design and build accordingly. Design for change, modifiability and adaptability.

The most damaging aspect of ignoring this rule is when people close their senses. Without a healthy dose of this rule people believe they know everything they need to know. Therefore they don't continue to ask questions or stay alert to new information. This attitude results in a crazy self fulfilling prophecy. If they're not looking they'll most likely not find any new information.

User Rule : The User Knows Only What They Know.

"Don't ask the blacksmith to race the horse."

Once we understand the "Know Rule" then it follows that the user doesn't know everything about the system either. Yet how many times have I heard... "Go see such and such... he/she knows <u>all</u> about the system." There are limitations to their knowledge, no matter how experienced they are or how many years they've been doing "it."

This rule means more than a pure application of the "Know Rule." One important point is that the user is so tied to what a system does now they generally can't see anything different. Additionally, they most likely don't see things from a system standpoint. It follows logically that most users will not be adept at designing quality systems. Therefore it is surprising that fads proposing that "users" do system design keeping popping up.

What is needed is to use the general system rules and other technical knowledge to look beyond the user's limited perspective. This can best be done by highly skilled and experienced system professionals.

Exception Rule : The User Will Only Tell You About What Usually Happens.

The user will only tell you about what usually happens not all the exceptions! Beware of this trap. No matter what they say it doesn't always work that way. Stated another way, when the user say's "always", watch out.

Systems need to be designed to handle the ordinary efficiently and the exceptions effectively. Plus, if one applies the "Change Rule" even if it does work a certain way today doesn't mean it will tomorrow. How many times have the "one" type of inventory (or employee, part, customer, etc.) turned into "two" or "three" different types?

Two Head's Rule : Two Heads Are Better Than One, Doesn't Mean Ten Are.

> *Terman's Law of Innovation: "If you want a track team to win the high jump, you find one person who can jump seven feet, not seven people who can jump one foot."*
>
> *"Too many cooks spoil the broth."*

When it comes to analysis and design, the optimum number of people is probably between two and four. Yet analysis-design meetings so often have eight or ten or more people. The impact of the knowledge and capabilities of the best people are diminished significantly in large groups. There is no way to prove or disprove most design decisions. In any large group, poor designs can result from poor proposals by well meaning but inexperienced and unknowledgeable people. They end up being accepted because they cannot be proved wrong. It sounds crazy but it is true.

The better way to include more people in the analysis-design is to discuss issues with many people individually. Then have larger group meetings be review meetings. The purpose of the latter is to catch things that may have been missed or promote team feeling not to design.

Importance Rule : Size Is Not An Indicator of Importance.

Greer's Law: "In systems the little things count"

Bionde's Law: "If your project doesn't, work check the part you didn't think was important"

Component size gives no indication of its importance. Or another way to say it there are no unimportant parts.

This interesting rule is at the heart of what makes the analysis-design of systems so challenging. In many other endeavors it may be possible to ignore significant numbers of small details. Those endeavors can still be successful. Any experienced system professional can think of countless examples of the smallest detail causing the result/product of the system to be useless. In systems, a single bit set wrong can negate a billions of bits set right.

Paradoxically, given the "Know Rule", which says you can't understand it all, is it any wonder that systems is such a challenging undertaking? Less cheerfully, is it any wonder that there are so many disasters?

This rule, as much as any rule, is one that gains tremendously from more experience. Added experience is the only way I know to develop the "second sense" for what small details are critical to any particular system design.

80/20 Rule : 20 Percent Of Anything Accounts for 80 Percent of the Result.

This is definitely a significant rule but needs to be balanced with the "Importance Rule." You can't assume in systems that because some 20 percent of the system does 80 percent of the work that the other 80 percent of the system is unimportant. This can be absurdly applied such as to say that since 80 percent of the cases in the accounts payable system are handled by 20 percent of the code, that we can ignore the other 80 percent of the code. Nevertheless, this rule has wide applicability in systems.

Forest and Trees Rule : Quality Design Is A Forest And Trees Issue.

"Losing sight of our goal we redoubled our efforts."

"The customer doesn't care about how, only what."

By forest and trees, I mean that one must be able to see both. Systems analyst-designers must be able to see both the big issues and the small issues. To ignore either will result in a lower quality system.

This may be, in part, why there are so few excellent system analyst-designers. Very few people are able to see at both resolutions. Rather, there seems to be either mostly detail, bits and byte people, or big picture people. In spite of this reality, everyone can improve by at least understanding this rule and consciously trying to focus on both levels.

UNIQUENESS RULES

Uniqueness Rule : No Two Systems Are Identical.

Beach's Law: "No two identical parts are alike."

The Uniqueness rule states: no two systems are identical. And if <u>any part</u> is different then the resulting behavior or results of the system <u>may</u> be different. If it worked last week, doesn't mean it will work this week if anything has changed. And something always changes, if only time. You see, time is a component of every system. Therefore, every minute that goes by potentially changes a system or its resulting product. If you are skeptical I hope you remember these words on January 1, 2001 when a high percentage of the computerized date routines stop working!

There is a more general form of this rule that says no two things or situations are identical. How many times have you heard someone say "We tried that last year and it didn't work." I would suggest that with the change of one single component, person, method, etc the same thing could be wildly successful. Be careful in applying this rule. Especially be careful in explaining your reasoning. This warning is because so much of "everyday" logic do understand the validity of this rule.

Alike Rule : Systems Are More Alike Than Different.

The concept that systems are more alike than different is at the heart of general system rules themselves. Without believing that systems have many things in common it would not be possible to propose valid system rules.

But I just made a case for the uniqueness of systems in the Uniqueness Rule. This seems to be a paradox. It really isn't one at all. Systems all follow general system rules. These rules are not about specific actions, performance or results. They are rather about the general actions, performance and results. It may be helpful to think in terms of the system called a person. People have many similarities, yet as individuals they are still unique.

SYSTEM MAINTENANCE RULES

System Life Rule : A System Dies Because Knowledge About It Diminishes Over Time.

> *The Computer Programmer's Lament: "Program complexity grows until it exceeds the capability of the programmer who must maintain it."*

Over time the knowledge about a system diminishes to the point where no one understands it well enough to make a successful fix or modification. Turnover is one of the reasons. Documentation another. The original designer and builders are the ones who knew the most about the system. Yet none of these people knew all the system. As time goes on these people leave or get reassigned. Each "generation" of people coming later to support the system know a decreasing amount about the original intent, workings and design trade-offs.

Understanding this rule helps in a number of ways. It increases our ability to plan for the real obsolescence of systems. It tunes us in to the decrease in quality of support that will occur over time. It increases our sensitivity to the importance of documentation. It also highlights the increasing energy\cost that is required as a system ages.

Documentation Rule : A Picture Is Worth A Thousand Words.

"This manual is useless!"

"The person who wrote this ought to be shot!"

How much documentation is not worth the paper on which it is written? I'd guess 90%. Why? Well, the written word alone is a terrible way to describe most systems. At least the written word without good accompanying pictures is. And most documentation relies far too heavily or exclusively on the written word. Additionally, most systems people are poor writers. The crazy illusion is that systems people think that what they've written is so much better than it really is.

I've studied the issue of what a system person does when trying to design a new system or understand an existing system. The best way of describing it is to say they are building a picture of the system in their mind. Any documentation pictures that help represent the system are extremely valuable in increasing the speed and level of comprehension at any phase of the system life cycle.

So, to improve things we need to take two steps. First use more pictures and diagrams. Second, use trained technical writers to produce the written documentation. Have them use the system developers and users as resources.

A final aside about documentation. I hear people say that something doesn't need to be documented because any fool should be able to understand it. The only fool is the person who believes this statement. There is no doubt that for every five minutes spent on good (ie graphics emphasized) documentation you'll save at least an hour later on.

Three Time Rule : To Understand, Read It Three Times.

"Manuals should be read like love letters; every word is important." Alfredo Rego.

This rule came in part from observing the very best systems people. They are great manual-documentation readers. They have learned that one reading of any manual is not enough to understand it. They will tell you the common story of having discovered something in a manual that they swear wasn't there on a previous reading.

Conversely, new graduates seem to have come through school believing that to understand anything requires only one reading or less. And if after reading it once they don't understand it then the solution is to go ask someone!

This is such a simple rule to understand. Figuring out how to get people to do it, isn't.

QUALITY RULES

Quality Rule : To Get Quality You Have To Define It.

"I'll Know It When I See It."

There is so much hype about quality these days. It sounds a lot like motherhood and apple-pie. The bottom line is that to achieve anything including "quality," you have to define what it is. It does little good to promote quality as an abstract objective because it is always judged in a concrete context. The Japanese have "discovered" the truth of this concept. They spend tremendous time researching what the consumer defines as "quality." That's why they produce products that consumers freely choose many times over American products.

Applying this directly to systems means that each component (speed, ease of use, defect rate "bugs", etc) making up a quality system needs to be defined. Further, the relative importance of each component quality definition must be weighted for importance and "essential" level. This needs to be done <u>before</u> the system is produced.

So the next time someone is going to develop something for you, ask them to define the quality characteristics it will have. It will be enlightening, I guarantee you.

Inspection Rule : You Can't Inspect Quality Into A Product, Rather It Must Be Built In.

This rule is very closely related to the Quality Rule. This rule does not mean that inspection should not be part of producing a quality system. Rather it means that if quality is well enough defined, and if each person knows what is expected of them, then they become their own inspector. They can build the quality in as they create the system. This technique has been shown, by the success of the Japanese in applying it, to be powerfully correct.

We need to spend far more time in systems figuring out how to build quality into our products.

PEOPLE RULES

Fallibility Rule : Everyone Makes Mistakes.

"To err is human, to really screw up takes a computer."

"Trust everyone but cut the cards." G. Weinberg

"In God we trust, everyone else we monitor"

There is no perfect person. So whether it's in the building or the using of systems plan accordingly. Design your method of developing and maintaining to make sure the mistakes are catchable and correctable or are immaterial. Designing for fallible users means many things. It includes editing adequately. It also means building as much logic checking into critical process steps as possible. Further, allowing for backing up and redoing things has to be considered. Documentation is also extremely important.

Las Vegas Rule: Never Bet On A Poor Performer Because You Think They're About to Improve.

Systems are more changeable than even people. The best predictor of a person's future behavior, performance and products is to look at their past.

Many times the best thing you can do for a poor performer is to fire them. Remember that a small change in a component of their environment may make them produce better. So it's very difficult to predict how they'll do somewhere else. This would almost argue for keeping them and trying to change different factors hoping to make things better. My view is that this is at best a long shot. We just don't know enough about humans and changing behavior to be successful more than a small fraction of the time.

Bottom-Line Rule : The Best Systems Are Written By The Best People.

Regardless of the tools, training and techniques... the best systems are written by the best people. Tools, training and techniques can't make a mediocre system person super.

Therefore, if you want the best systems, hire, reward and retain the best people. That is why this is the bottom line rule.

One way to do this; figure out how to pay the 20% of the people who do 80% of the work closer to 80% of the pay!

SUMMARY

We in systems have been operating within the bounds of system rules. They do exist. Let's take the time to figure out what they are and apply them. More importantly let us teach these rules to the beginners. There is much to be gained by doing so.

$pending to $ave on $oftware

presented by

Shelby Robert
General Manager
Proactive Systems
Four Main Street, Suite 101
Los Altos, CA 94022
(415)949-9100

A. Introduction

As this paper is written, things look pretty grim for the economy. Many organizations are facing budget reductions and cutbacks in spending that are unprecedented in the memory of an entire generation of MIS people. Times are tough and so are the managers who control the pursestrings.

Still there are times when a purchase of software really will save the company real dollars in terms of reduced costs, or in terms of increased productivity. So the question facing the MIS manager becomes "How can I demonstrate to my management that spending a few dollars now will help my company to do better in these troubled times?"

This is the question that must be answered after you have completed a thorough technical evaluation. The product has been selected. You're satisfied that you have the *right stuff* in hand, but now it has to be justified to top management. It is necessary to convince them that the decision to purchase now is the best thing that the company can do to move itself forward.

In the end this paper is about developing numbers that support your decision to move forward with a purchase. Now we approach a different task. The job is measuring the relative value of making this purchase as opposed to not making it. How positive will be the effect on the company of making this purchase?

B. The VALUE Equation

The real question to be answered in looking at acquiring software is *What value will my company derive out of this purchase?* If the answer is nothing, then we might as well stop right here. If, on the other hand, you perceive that there is real value for you and your company, the balance of this paper is designed to help you measure that value.

So how do you measure value? One definition that seems to hold up pretty well is:

VALUE = BENEFITS - COST

We will refer to this as the "Value Equation." It is obvious that as measured benefits rise, and as the costs decline, the value of the purchase to your company increases. The greater the difference you can show between the benefits of acquiring an item and its cost, the more convincing and persuasive your presentations will be. Because the price of your proposed acquisition is fairly obvious at this stage, the real key to understanding the power of the Value Equation is in measuring benefits.

C. Identifying Benefit Areas

Measurable dollar benefits for the MIS department generally come in one of four basic varieties:

a. Reduction of materials costs
b. Increased MIS Staff productivity
c. Increased User Staff productivity
d. Reduced space/power/support services consumption

Individual software purchases will not necessarily find benefits in each of these areas, but most of the major measurable benefits will fall into one of them. At this point, an example may help to illustrate these issues.

One of the issues facing many MIS departments today is the question of replacing preprinted forms with electronic ones. Let's examine some benefits of this way of doing business that touch on each of the four areas mentioned above.

So that we are on some common ground, electronic forms are forms that are printed at the same time as your data on a LaserJet printer. The obvious advantage is that there is no need to have a supply of preprinted forms on hand in order to print such things as invoices, purchase orders, packing lists, and other documents. They simply print on plain white paper, including your logo, with terms and conditions on the back, at the selected printer, and with the appropriate number of copies. All of this sounds great, but what are the measurable benefits here?

Benefit #1. Cost of the forms.

The first and most obvious benefit is that the cost of the preprinted form goes away. But this paper is about measuring things, so let's play with some numbers. Suppose you run a manufacturing company's MIS department and each working day you print 500 sets of four-part invoices that cost you $0.20 per set. At 250 working days per year, your annual consumption is 125,000 for a total annual cost of $25,000.

Unfortunately, there are some materials costs associated with LaserJet printing which must also be factored in here. The toner cost per face is about $0.015, and since our invoice requires terms and conditions on the back of the original, we will have five faces to print. In addition, the paper itself costs about $0.004 cents per sheet, and we need four sheets. So our total costs to produce the same form as above are:

125,000 * (($0.004 * 4) + ($0.015 * 5)) = $11,375.00

To determine the real measurable benefit to the company, we now subtract the old costs without our software acquisition from the projected new costs with the software in place to find the dollar amount of the benefit.

$25,000.00 - $11,375.00 = **$13,625.00 per year**

So far in examining this benefit we have looked at replacing only a single form. If you were actually contemplating such a purchase you would list the savings to the company for each of them. The formulas remain the same, only the names are changed to protect the expensive. So let's explore one more just for good measure.

Another form that you probably deal with is a monthly statement. Let's assume that our fictional company has 1,000 active customers, and that the statement is in two parts, one for the customer, one for the Accounts Receivable department. Using the same numbers as above, the preprinted set will cost about $0.10 per set. Printing 12,000 per year results in an annual paper cost of $1,200.00.

Again using the same costs as above, two sheets of paper at $0.004 each and three printed faces with a toner cost of $0.015 per face yields a set cost of $0.053 per set. At an annual cost of $636.00, this is still a savings of $0.047 per set or $564.00 over the 12,000 printed in a year.

$1,200.00 - $636.00 = **$564.00 per year**

Just a word here about presentation. All of us like to have information about which we are being asked to make decisions presented in an easy to follow, understandable way. Although the supporting details are important to reveal thoroughness in the discovery process, the major points are the ones that need to be highlighted. It is nice for the reviewer to have available all of the details, but he will appreciate a presentation that helps the important numbers jump out so that his time and energy are saved the task of extricating them from the "Spreadsheet Sea." One of the ways to improve presentations is to tabulate results.

If you have several forms to deal with, you might want to summarize in a format that looks something like this:

Form Name	Sheets per Set	Savings per Set	Annual Usage	Annual Savings
Invoices	4	$0.109	125,000	$13,625.00
Statements	2	$0.047	12,000	$564.00

Total Annual Savings for Benefit #1 =--> $14,189.00

Benefit #2. MIS Staff Overhead Reductions

The MIS staff have a built-in burden in handling pre-printed forms. When forms printing is needed, the forms must be mounted on the printer, printed, decollated, burst, and distributed. This is apart from a weekly inventory, and the involved process of reordering when quantities run low. Consider the following numbers for our sample manufacturing company:

Operations time (daily):

a. Printing forms	1 hour
b. Decollate / Burst	1 hour
c. Distribution	1 hour

A total of three hours per day is 15 hours per week times 52 weeks totals to 780 hours per year. Plus additional overhead of one hour per week for the weekly inventory. This leaves us with a total MIS operations overhead of 832 hours. There is a lot of training and investment in MIS operations people. The cost rate for them is relatively high in comparison to office or plant workers. A rate of $50.00 is not unreasonable in figuring total costs for the skill level and responsibility required of these folks.

All of the above overhead in the MIS department can go away entirely with electronic forms, so all of the costs are either direct savings, or opportunities for MIS staff to be productive doing other things. Either way the benefit to the company is:

832 hours * $50.00 = $41,600.00

Every time a form must be reordered, it is reviewed by management. If not only by MIS management, then by other managers in the company. The review must be painstaking because reorder points are infrequent and volume purchases, necessary to get the unit price reasonable, are still quite expensive. To get away with reviewing a form in four hours seems quite reasonable. The people who must do the final review are sophisticated management people. Their hourly cost is reasonably in the $100.00 per hour range. So with 20 forms to review per year our current cost equation is:

20 forms * 4 hours per form * $100.00 per hour = $8,000.00 per year

Because electronic forms can be changed on a moment's notice, lengthy regular periodic reviews are not required. Some overhead will still be necessary, but reductions of 75% are not unusual. This still leaves us with a cost savings of $6,000.00.

The total of reductions in MIS overhead is now the combination of the two numbers from above.

Benefit #2 = $41,600.00 + $6,000.00 = $47,600.00

Benefit #3. Increase in User Productivity

No MIS department survives in a vacuum. MIS embodies the very definition of a services business. We have no reason to exist without our users. To go one step further, it is our users who are doing the business of the company, not those of us in MIS. If we can contribute to the productivity of our users, we will have a direct beneficial effect on the company's ability to do profitable business. One of the ways that we do that is by reducing the time it takes us to deliver information to our users.

In using pre-printed forms, we wait and batch things like invoices until we have an opportunity to mount them on the printer. If the picking list is the fourth copy of that invoice, the goods can't go out to the customer until the invoice is printed. The third copy of the invoice must be carried to accounting for their needs, and the original and remittance advice will go to the mail room to be posted.

With the introduction of low cost LaserJet printers and the availability of software to drive them, we can now print the original and remittance advice in the mail room, the picking list in the warehouse where it is needed, and the accounting copy in the accounting department, all without any delays. The overall result is that needed information arrives at its destination where it can be used in a more timely fashion. Productivity will rise due to increased efficiency of delivery of information.

To illustrate how powerful productivity gains can be, let's assume that our manufacturing company has five people in accounting and ten in the warehouse. If we can realize a modest 1% productivity increase for those people by improving information delivery, we will save 6 hours of labor per week that can be used for accomplishing other tasks.

At $20.00 per hour average cost to the company to have an employee on staff and working, the annual productivity gain to the company is:

Benefit #3 = $20.00 per hour * 6 hours per week * 52 weeks = **$6,240.00**

Benefit #4. Reduction of Space

The 20 forms that we have been using must be individually stored and inventoried each week. They use floor space in the computer room for the working store. In the storage area which is secured, they use additional space.

Computer room space is expensive at about $10.00/sqft per month. At only one square foot per form, the cost is $200.00 per month, just to store forms. In the storage area the cost of $3.00/sqft per month is less, but each form will require an average of 10 square feet. This amounts to $30.00 per form per month or $600.00 per month in the storage area.

Again, the total will not be reclaimed, but at 75% savings (because we now will stock only white paper) we will reclaim $600.00 per month of floorspace. The annual total is:

Benefit #4 = $600.00 per month * 12 months = $7,200.00

D. The Bottom Line

The benefits listed above are not an exhaustive list by any stretch, but they serve to illustrate how to identify and measure the benefits that will accrue to your company by making such an investment. Now we can put it all together for your management.

First, a brief review of the benefits that we have identified and their dollar value to the company.

Benefit #1. Direct Forms Cost Savings $14,189.00

Benefit #2. Increased MIS Productivity $47,600.00

Benefit #3. Increased User Productivity $6,240.00

Benefit #4. Reduced Space Requirements $7,200.00

Total Annual Benefits $75,229.00

All of this does assume the purchase of a software package to do all of these marvelous things, and the purchase of some LaserJet printers to do the work. The discussions above have mentioned a printer in the warehouse, a printer in the accounting department, and one in the mailroom. MIS will of course have to have a printer to do their testing. Assuming that all of the printers purchased were LaserJet IIIDs, the total investment in printers would be about $8,000.00. The software to do the job above will cost an average company about $7,000.00.

These two numbers represent the total capital investment for the job. The problem is that comparing the $15,000.00 total of these two investments to our benefits list is like comparing apples and oranges, since our benefits are expressed as annual amounts, but the benefit from the investments will last longer than one year.

The best way to evaluate the viability of a capital investment is to use a discounted cash flow methodology. A discussion of this method is a little beyond the scope of this paper. Another, simpler approach is to look at the amortized cost over the life of the investment and the payback period.

Capital investments must be amortized over their "useful life." The tax man defines useful life for computing investments as being five years. What this means is that only one fifth of the cost of the hardware and software should be taken as an expense in any single year. In our example, this represents an annual capital cost for our project of only $3,000.00.

Notice also that the benefit numbers above are all annual numbers. That means that these benefits will accrue to the company each year of use. So now a look at our bottom line numbers:

Total Annual Benefits $74,665.00

Total Annual Capital Cost $3,000

All of the things we have mentioned above are important. They are the essence of what must be communicated to your management, but so far they are only numbers. For many people, numbers don't have a life of their own. They just sit there. In reality, numbers by themselves are just plain

Boring!!!

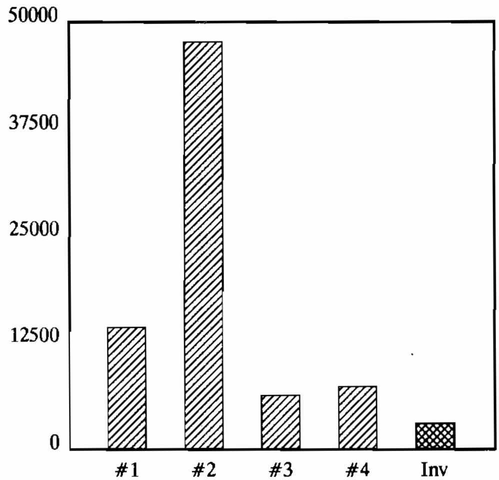
For you as an MIS Manager, this means that you have to do something to get your management's

ATTENTION!!!

How can you do that? If the problem is that all you have to work with is numbers, why not paint your management a picture of what the numbers mean. Pictures of numbers in graphic form convey meaning instantly, intuitively, without having to do the mental work of internalizing the specific value of each benefit. By painting management a picture, you save time, help them to make a better (as well as faster) decision, and make it easier for them to accept your arguments in the process. In the process you don't do any harm to your own reputation for professionalism.

What might a picture of our results look like? Let's take a look at the effect of illustrating the relative value of each of our benefits and our annual investment.

Individual Benefits vs. Investment
Annual Amounts



By looking at the above chart, it is obvious immediately that any one of our four benefits would be sufficient to justify the purchase. The combination of the four is an obvious and overwhelming argument.

Let's go back now and take another look at the VALUE EQUATION. The VALUE EQUATION above stated that VALUE = BENEFITS - COSTS. Solving the equation with the numbers we have developed here yields an ANNUAL VALUE to the company of $69,865.00. Over the five year life the tax man requires, this investment will return $349,325.00. Even the most callous of CFOs will take note of a return like that.

E. Time to Return

One of the calculations that you will want to make has to do with the time to return the investment you are asking your company to make. The easiest way to do this is on a monthly basis. First, divide the annual benefit by 12 to find the monthly benefit.

$75,229 / 12 = $6,269

Now divide the total investment by the monthly benefit to find the number of months it will take to recover the total invested.

$15,000 / $6269 = 2.39 months

Again a very favorable number. This company cannot afford to wait another day for this purchase.

F. Summary

In the preceding pages we have examined only one software purchase, that of a LaserJet printing package. We have certainly not examined all of the benefits that might be associated with such a purchase. What we have examined is a methodology for quantifying the benefits to be derived by a software purchase.

The basic benefits that your company will derive from a software purchase will generally come from one of the four sources mentioned above: reduced materials cost, increased MIS staff productivity, increased user staff productivity, or reduced consumption of other company resources. In order to justify such a purchase to management, the individual benefits need to be itemized, and quantified into dollar savings or productivity gains.

Once the benefits are identified, quantified, and valued, a little care and attention to the presentation of the information goes a long way with management. If you are convinced that this is the right decision for your company, it is worth spending a little time to make it easy for your management to understand your position.

The numbers that you generate in preparing a cost justification may yield totals that seem unreal. Your benefit amount may be dismissed as not being a hard dollar savings. The reason is that your company is probably not going to fire anyone to reduce their costs as a direct result of this sofware purchase. Nevertheless, the arguments for the benefits still hold because the time saved can be put into more productive work.

The important issue in justifying software is that the benefits to your company must be itemized and valued in order to demonstrate to your management that the acquisition you propose is the right thing for the company to do, and that this is the right time to take action. By going through the exercise of identifying and quantifying the benefits of your purchases, you will strengthen your bargaining position and your case for the purchases that you need to make to help your company function better.

This article was typeset using FANTASIA from Proactive Systems.

MOTIVATING YOURSELF AND YOUR STAFF

Lynn A. Novo, CPC
Network Systems Company
10 Diana Court
Cheshire, CT 06410
203-250-1919

When has employee motivation been more important than in these uncertain economic times? How can you as a manager expect to motivate others when your own job security may be in jeopardy? If money is the primary motivator, how can you improve productivity when salaries have been frozen or bonuses have been eliminated?

It's time to get back to basics. Let's exchange ideas that will get you thinking about creative ways to motivate your team of HP professionals. This is not a discussion of motivational theory. Instead, we'll apply that theory in practical ways that can help your department. We'll talk about how important you, the manager, are in the process and highlight several effective self-motivators. The carrot, stick, and soul food methods of motivating others will be outlined with specific examples of each for your HP staff. We will discuss effective low cost motivators and mention common demotivators for HP professionals. We'll end with some valuable resources to help you long after you leave this session.

MOTIVATION BEGINS WITH YOU

Like so many things that impact our success as managers, motivation begins with us. How can you possibly motivate others unless you are motivated, alert, and enthusiastic yourself? If you demonstrate a lack of commitment or interest in your people and your job, your staff will notice. Why should they bother if their boss doesn't?

The myriad of self-help books, journals, and video tapes on the market today can bombard you with information on how to motivate yourself. At the core of this information is how to create an environment for yourself which allows natural self motivation.

* **Self evaluation:** You must first assess where you are on the motivation spectrum before you can improve. This is the hard part: stand back from yourself, be critical and honestly evaluate your present situation. What do you really think of your job? The people you work for? The people who work for you? How do these feelings impact your workplace?

Have you ever asked a colleague to critique you? You may not realize how you are coming across to your staff. Do they feel that you are a part of their team, or are you an outsider? Your motivating intentions may be thwarted by some non verbal cue you're not even aware you're giving.

It is also very effective to ask the people who report to you to evaluate you as a manager. Have you ever asked someone, "How am I doing as a manager? What could I do to improve morale in the department?" You will probably be surprised by the candor of the answers.

* **Starting the day right:** Have you ever noticed that when the day starts badly, it somehow never improves? It is really tough to be a motivating manager if you overslept, didn't get a cup of coffee, and got caught in traffic on the way into work. No matter how hard we try, there always will be some days like that. But to the extent that we can control our early morning and get the day started with organization (and even serenity if that's possible), the better motivators we will be.

Early Morning Primers

- Do your best to be physically and mentally alert by the start of the work day.
- Eat breakfast.
- Schedule some exercise, even if it is only a brisk walk

through the parking lot.

- Wear clothes that make you feel good.
- Don't dwell on all the negative aspects of your job or your problem employees on the way in to work.
- Do something you enjoy before arriving at work.

Are any of you morning people? What do you do to motivate yourself in the morning?

Some ideas:

Read the newspaper over a cup of coffee before you leave the house.

Telephone a close friend or relative for an early morning chat.

Watch TV or listen to your favorite music.

Do a small task around the house that you've been putting off.

Take the "scenic route" into work.

Everyone has an "off-day" once in a while, but as a manager, you must force yourself to be positive. Before leaving the house, look in a mirror. What do you see? That is what will greet your staff.

It's all in the entrance -- How does your boss walk into the office in the morning?

* **Style of entrance:** Greet your staff by name and with a smile. Chat informally and let them know you are interested in their life outside work. Create positive energy and enthusiasm that will infect your staff throughout the day.

A programmer mentioned to me how her boss, who was usually bad-tempered and unapproachable in the morning, came in one day with a smile on his face and stopped at each of the workstations on the way to his office to greet his staff. The atmosphere in the department that day was decidedly different-- a little confused, perhaps, but optimistic. When the manager entered the office with the same energy the next day and the next, a real change occurred. Solutions to problems seemed to materialize out of no where and new, creative ideas were generated. The whole place seemed to buzz with energy. Everyone's attitude about their jobs improved. My hope is that the manager learned a lesson and continued to make his style of entrance a motivating one.

* **Personality:** Work hard to be perceived as a good listener. Make yourself available to your staff so they know you as a person, not just the signature on their annual review.

It is easier to keep a positive attitude if you do not dwell on negatives. This sounds simple enough, but don't we in fact think more about the projects that are overdue than those that were completed ahead of schedule? And don't we think more about the problem employee than the one that's cranking out work at a record pace? As managers, we need to try very hard to keep a perspective between the good and the bad at our workplace. Concentrating on the negatives will have a demotivating effect on us and our staff.

We surveyed HP professionals and asked them to profile the personality traits of the most motivating managers they had worked for. What are the traits you have found motivating?

Good communicator- expresses clearly what, when, and why something needs to be done.

Flexible- willing to bend when necessary. Not afraid to admit they were wrong.

Defender- stands by IS team member and serves as "insulator" from the heat of upper management and users.

Reliable and consistent- establishes standards and keeps them; consequences of wrongdoing are predictable.

Good listener- really hears what is being said, and remembers it.

Perhaps the most important trait is:

Realistic- sets achievable goals and deadlines for projects. Builds in contingency plans whenever possible.

Managers who are strong in these personality traits are able to engender trust, respect, and commitment among their staff. From this foundation, a manager will have an easier time motivating others.

THE TWO H'S OF MOTIVATION

At the core of motivating others are two words not often associated with motivation. They are what I call the two h's of motivating: HONESTY and HUMOR. The managers we interviewed admit these concepts underlie all they do for their staff motivation. It is not so much a separate management technique as a fundamental way of life for these managers.

Honesty means calling the shots the way they are. Your staff has a right to know "the big picture" and the more often you let them see it, the more understanding they will be of the pressures on you. If your staff knows the commitment you made to get a project done, they will help you meet the goal. As IS professionals, they know that there will always be "gotya's" and unknowns, but if you are doing things that make sense, they'll go the extra mile.

Many managers overlook humor as a motivator. Humor relieves stress (you don't have any of that in your department, do you?), diffuses anger and hostility, and alleviates potential conflict.

What are some ways you can introduce humor into your workplace?

> Have a "joke of the day". Each department member (including you) brings in a cartoon or joke and posts it where everyone can see it.
>
> Begin each meeting with a humorous quote. (The side benefit of this is that it will help your staff "shift gears" from whatever they were working on before the meeting.)
>
> Have a "stand up" meeting for the jokester in your department to tell a funny story to the group.

Include a funny icon or cartoon on your next interoffice memo.

Any of these ways will get your staff thinking about humor as an acceptable part of your workplace. One caveat though-- be careful not to make the users the brunt of all your jokes. Your overall objective as a manager is to foster good relations between departments. Make sure you're not making a laughing stock of your key user, tempting as that may be.

As we leave motivating yourself, please remember that there is a fine line between being buddy buddy with your staff and representing management. I am suggesting that you be approachable and real. You alone need to determine how much time outside of the office is appropriate to spend with your staff members.

MOTIVATING OTHERS

We've been talking about motivating yourself as a manager. Now let's discuss motivating others.

The instruments of motivation can be classed as the rewards, punishments, and needs satisfaction. What do you immediately think of when someone mentions motivation? Something tangible like money? Or the threat of losing your job? (I know plenty of high paid MIS

Directors who have this motivator.) What about an intangible such as a promotion or greater responsibility?

As a manager, you should know all the motivating resources available to you. Some of them will depend on your company's policy and your budget, but many more will rely upon your skill as a motivating manager. The important thing is to utilize a variety of motivators and recognize that different tools will work for different individuals on your staff.

Unfortunately, there is often a great disparity between what a manager thinks is a motivator and what actually does motivate. So Step One is identifying the personality traits and "hot buttons" of the people on your staff.

Are IS People Different?

Two researchers by the names of Couger and Zawacki carried out an extensive study of computer personnel and identified several personality characteristics. Their findings were published in the book, Motivating and Managing Computer Personnel.

They identified a "typical" computer person as displaying the following traits:

- high growth need (increasing skills and responsibility)
- high quality performance (pride in what they produce)
- high need to achieve (drive to do well)
- low absenteeism (commitment to work the job, not the hours)

Stereo types are dangerous and it would be unfair to assume that every member of your IS staff possessed these traits. As a manager you must take each of your staff members as an individual and recognize his or her needs, drives, and motives in order to develop a personalized motivation strategy. But using Couger and Zawacki's model as a reference, what things can motivate IS professionals?

1. understanding and listening by the boss
2. competition with others - have a reward as a prize
3. travel incentives
4. recognition from other departments or upper management
5. salary raise
6. salary cut or no raise
7. change in job title
8. responsibility for equipment
9. time available for learning
10. gifts at holiday times

11. progressive career ladder within the organization
12. company stock offering
13. working as a team on a project
14. above average benefit package
15. tuition reimbursement
16. work area moved closer to boss
18. supportive coworkers
19. lunch with the boss
20. dress code flexibility
21. job title
22. management openness and honesty
23. flexible work hours
24. cash bonuses
25. praise - open appreciation for a job well done
26. personalized attention by the boss
27. special assignments, pet projects
28. maintenance assignments instead of new projects
29. private office or nice location in office
30. administrative support (staff or office equipment)
31. access to classified data
32. designated parking place
33. Friday pizza party
34. company sports teams
35. fitness facility on premises or health club membership

36. comp. time

37. participation in conferences or user meetings

38. support of CSL or trade publication submission

39. good personal relationship with superior

40. challenging assignments

41. task variety

42. ability to see task to conclusion

43. promotion

44. demotion

Would everyone on your staff be motivated by all of these things? NO! The point is to know the full range of possible motivators and select the ones that are appropriate for a given individual at a given time.

CARROT, STICK AND SOUL FOOD

The instruments of motivation can be classified as rewards, punishments, and needs satisfaction. What do you immediately think of when someone mentions motivation? Usually something tangible - more money, promotion, threat of losing your job etc.

As a manager you will be aware of the resources available to you, but it is the skill you apply in choosing them that will make these motivational vehicles effective.

Unfortunately, sometimes there is a disparity between what a manager thinks is a motivator and what actually does motivate his staff. So before administering any motivator, try to ascertain the most effective one for that individual. No point giving a programmer a 15% raise when what he really needs is a job title change to programmer analyst and involvement with users.

Successful managers will use a combination of carrot, stick, and soul food techniques to keep their staff highly productive and motivated.

CARROTS (also known as incentives, rewards or perks)

These are traditional means of motivation and are usually recognized as tangible objects such as: wage increases, cash bonuses, benefit packages, company car, trips or other incentives.

Carrot incentives are used on a short-term basis to increase enthusiasm and bring immediate benefits. If sources are available they are easy to administer. The drawback is that they can fail to provide long-term effects. Over time incentives get taken for granted, and often must be increased or improved to remain effective.

STICKS (also recognized as threat, fear or punishment)

Sticks are easiest to remember as being the opposite of what we generally think of as motivators. But the negative is just as powerful a motivator as the positives we outlined. You may not realize it, but the threat of being assigned the most boring maintenance project may in fact be what's motivating some of your staff members to get their work done on time. In the face of threat, most professionals will respond by getting the job done and on time.

Like carrots, sticks achieve short-term productivity. They work best in small doses and when the reason for their use is clearly defined.

Both carrot and stick techniques are reinforcement methods, which over time can lose their impact. Both the rewards and the punishments need to be increased to remain effective. You can see how this could become impossible to maintain. A more viable and effective motivator, especially for computer personnel, is to satisfy inner needs via the soul food method.

SOUL FOOD

IS people are big soul food eaters and tend to be more motivated by inner needs satisfaction. These tend to be intangible and encompass:

- intellectual challenge

- sense of purpose
- more flexibility and responsibility
- greater freedom and recognition
- increased input into decision making

These are the things that give people the warm fuzzies about their work. Unfortunately, it's difficult for many managers to identify soul food motivators because they tend to be intangible. Nonetheless, they are real and can be used effectively.

Let's revisit Couger and Zawacki, whom we mentioned earlier. Robert Zawacki has said:

> "More than 50 percent of computer employees' motivation and productivity depends on how well they have been matched with their jobs."

Couger and Zawacki surveyed more than 6000 data processing professionals and identified five core job dimensions that are required for job satisfaction. They are:

1. Skill Variety - use of different skills and talents to complete a project.

2. Task variety - degree to which the job requires completion of an identifiable segment; i.e. doing a job from beginning to end with a visible outcome.

3. Task significance - extent to which the job affects other people, both in the company and the external environment.

4. Autonomy - degree to which the job provides independence and the individual's discretion in scheduling the work and determining procedures.

5. Feedback from the job - obtaining direct and clear information about performance.

As you can see, one of the biggest "soul food" motivators for computer personnel is the job itself. Regular assessment of each of Zawacki's job dimensions should guide you in providing the soul food motivators for your staff.

Here is one example of a soul food motivator that may apply to someone on your staff. Many IS professionals have a low social need-- they do not need to interact with people or build relationships to be motivated at work. But this can be a dangerous generalization. If you have someone on your staff who is a "people person", you must recognize that building relationships and having frequent contact with a variety of people is important to that individual's job satisfaction. To motivate that type, you'll want him or her designated as the "user liaison" person, or key user contact person for specific applications.

The soul food method of motivating calls on your ability to recognize the different personality types among your staff members. Allocate time to get together with your staff to jointly set realistic goals, know who they are and what they need from their job. Then do your best to give them the intangible "soul food" so that they will thrive in your environment.

MOTIVATION ON A SHOESTRING

Many soul food motivators work well in difficult economic times since they often do not cost much. As we have said, money is one motivator, but it is not everything. The good news for motivating in these conservative times is that more attention is given to what the employee really desires or needs. Can we think of any "Motivators on a Shoestring"?

* Flex time
* Relaxed dress code
* Certificate of appreciation
* "Programmer of the Week" Award

* Popcorn

* Pizza party or group lunch (Dutch treat)

* Joke of the Day board

SOME COMMON DEMOTIVATORS

I shouldn't leave the topic of motivation without sharing the most common demotivators my colleagues and I have heard. If you recognize any of these, I hope you can correct or improve them to minimize their demotivating effect.

1. Physical environment is not supposed to be important to IS professionals. However, we have heard many complaints about how an unorganized work space is demotivating. Do you have lots of equipment boxes in the hallway?

2. The loss of challenge from the job or diminishing responsibilities will cause lack of motivation.

3. The inability to learn new skills and techniques will cause frustration and apathy. Keep your IS staff stimulated with new technology.

4. Insufficient feedback or only negative feedback can be demotivating. I like to "catch them doing something right" as a good One Minute Manager should.

5. Unfair or unevenly applied rewards kill motivation. Be consistent with all your staff members, so that no one is perceived as your "pet".

CONCLUSION

We have just scratched the surface on this topic but hopefully it will help each of us think about creative ways to motivate our staff. Motivation begins with us, so the way you start each work day and the attitudes you bring into your workplace are important. When you try to motivate others, remember your whole menu of motivating methods: carrots, sticks, and soul food. Recognize that individuals on your staff will need different things to be motivated. I hope I've motivated you to try a few of these ideas. Happy motivating!

REFERENCES

Bennett, Amanda. "When Money Is Tight, Bosses Scramble For Other Ways to Motivate the Troops." Wall Street Journal. October 31, 1990, p. B1.

Couger, Daniel, J. "Motivating IS Personnel." Datamation, September 15, 1988, pp. 59-64.

Couger, Daniel J. and Zawacki, R.A. Motivating and Managing Computer Personnel. New York: Wiley Interscience, 1980.

Couger, Daniel J., Zawacki, R.A. and Oppermann, Edward. "Motivation Levels of MIS Managers Versus Those of Their Employees." MIS Quarterly, September, 1979, pp. 47-56.

Mella, Michael. "Computer People: Yes, They Really Are Different." Business Week, February 20, 1984, pp. 66-77.

Von Oech, Roger. A Whack On The Side Of The Head. New York: Warner Books Inc., 1983

Wagel, William H. "Make Their Day - The Noncash Way!" Personnel, May 1990, pp. 41-44.

Weber, Joseph. "Farewell, Fast Track." Business Week, December 10, 1990, pp. 192-200.

Yate, Martin. Keeping The Best. Holbrook: Bob Adams, Inc., 1991.

Zawacki, Robert A. "The Real Key To MIS Motivation Is In The Matchups." Business Month, March 1989, pp. 77-78.

Zawacki, Robert A. and Couger, Daniel, J. "What Motivatates DP Professionals?" Datamation, September, 1978, pp. 116-123.

Zawacki, Robert A. "How To Keep Eagles." Computerworld, July 27, 1987, pp. 51-57.

PAPER NO.: 2807

TITLE: How You Can Get Your Employees To Pull Together Through "Interactive Management"

AUTHOR: David L. Sugg
Performance Contracting Group, Inc.
5435 77 Ctr. Drive #10
Charlotte, NC 28027
704-529-8034

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

Paper #2809
Risk Taking... In Your Career and Personal Life

Kathy McKittrick
McKittrick Associates
5547 South Yampa Street
Aurora, Colorado 80015

303-690-1550

Risk Taking... In Your Career and Personal Life

Risk taking. We've all done it. Sometimes we've won... sometimes we've lost. But we've almost always learned from it, and its more a part of our lives than we might think.

Risk is defined as "the possibility of harm or loss", but the term is more often used to describe actions that result in unpredictable consequences; positive or negative. And that's what we're talking about in the context of this paper.

For example, when we change jobs we take a risk. But before taking that risk, we calculate that there is a high probability that our new job will be more satisfying than our current job. This doesn't necessarily mean that the new job will meet our expectations. It simply means that we enter into it believing that the odds are in our favor, based on the data available.

How do we learn to evaluate risk? When we begin life, the world is full of risks. We can't avoid them, so we move forward with innocent gusto, forging ahead even after the bumps and bruises that come with new discoveries.

Consider the toddler who wanders away from his mother in a department store. All those bright colors, noises and possibilities draw him away before his small brain becomes aware of the possible consequences; losing his mother. When he realizes that she's nowhere to be seen, he panics, and the experience becomes a very unpleasant one. Even so, he's gained something; the knowledge that a world beyond his mother exists, the exhilaration of freedom and the first seeds of a self-confidence that will someday allow him to leave home.

As we continue to grow, the risks we take grow and change. By the time we reach our teen years, we begin to grow cautious in certain areas. We avoid the risk of feeling foolish in front of our peers, and yet at the same time, we risk the wrath of our parents almost daily (or at least every weekend) by indulging in behaviors that we know they won't approve of.

As young adults, we are bombarded with risky situations. In some we are at emotional risk. The teenage boy who calls a girl to ask for a date is at severe emotional risk, as evidenced by the knot in his stomach and the cracking of his voice as he asks. But he calculates his risk, ultimately deciding that the possible benefits outweigh the possible rejection. If he chooses never to take such a risk, he might end up alone.

When we go through the process of interviewing for our first job, we are called upon to behave in ways which are foreign and risky to us. Dressed in clothes that we wouldn't have been caught dead in on a college campus, we smile and shake hands and second guess what it might be that the interviewer wants to hear. The reward? A job. The alternative? Unemployment.

We've all heard the story about the aspiring actor who risks it all by barging into the producers office and demanding an audition. Or the one about the young athlete who crashes a major league practice to show his stuff. These young men have nothing to lose, and everything to gain, and so the risk becomes palatable.

That's part of the reason that we so willingly face risk in our early lives. Without risk there can be no movement or growth. We've got little to lose and everything to gain. Also, during our youth we don't calculate our risks very well, if at all. Our lack of experience allows us to take risks that later we wouldn't consider because we know what most of the variables are.

Somewhere along the line, as we mature, we begin to avoid risk. Risk is often uncomfortable, and the more comfortable our lives become, the less we want to "rock the boat".

There are a number of reasons for this. First and foremost is the "if-its-not-broken-don't-fix-it" mentality. If things are going along smoothly, why take a chance that an alternative will cause hardship?

Another important reason that we risk less as we age is that we become accustomed to the self-satisfying feeling of being experienced and knowledgeable. By the age of thirty or forty, we realize that we've spent much of our energy trying to eliminate risk from our lives. We marry, settle down in a specific job or career, order the same entree when we go to our favorite restaurant, and return to the same vacation spot year after year. Why introduce unknowns into our lives when we're comfortable and secure? Because the potential rewards can be great.

Consider Jimmy Conners. Last August, this thirty-nine year old world champion tennis player took an enormous emotional and professional risk when he played in the U.S. Open Tennis Tournament. He certainly didn't need the money. He could have made that much and more by playing the role of the aging athlete announcing tennis matches and making commercials. Instead, he trained and played all year, and gave it a try. What was the worst that could have happened? A humiliating defeat (such as the one John McInroe suffered) in an early round. This was no small risk for a man who had been at the top of his field.

He didn't win the U.S. Open. But he did make it to the semi-finals, and won the praise and admiration of the public and his peers. You could feel his exhilaration and heightened sense of self-esteem through the T.V. screen.

Another example is a retiring airline pilot who, at the age of sixty, decided to learn to ski. He considered the benefits; exercise, an activity that he could share with his children and grandchildren, and the self-satisfaction associated with accomplishing such a feat. He considered the downside; the possibility of broken bones, the grueling hard work involved in learning to ski, and the ridicule of other skiers watching an older man inching down the mountain. Today this man is seventy years old and has enjoyed ten years of skiing. During that time he's taken ski trips each year with his children and grandchildren, maintained his health, and gone on to enjoy other sports for which he never had time during his career.

These are two examples of people who have risked and won; both when they were well past what many of us may consider to be our "risk taking years". There are many reasons that they were successful; luck among them. But in addition, they had two things in common. They <u>believed</u> that they would be successful, and they calculated their risk and decided that the possibility of a positive outcome outweighed the possibility of a negative outcome. They also happened to be people who were more accustomed to taking risks most of us.

We know that there can be benefits to taking a chance. But how do we become more comfortable with doing so?

First, we can look at our history. We've all taken risks at one time or another, and its helpful to look back and review their outcomes. Figure I shows the risk-history of a forty year old woman. Its not an unusual history. Early in life, her risks were controlled by her parents. As she grew older and became more and more responsible for her own actions, her risks increased, peaking in her early twenties. As she continued to age, and presumably realized some success in her life, she became increasingly self-satisfied and reduced the frequency and number of risks that she took.

Figure II, a more detailed chart of her career risks, which is quite different from her overall risk chart, shows that her risk taking increased into her mid-thirties. According to the chart, she's due for her next big risk, but may choose, as so many do, to avoid it.

The purpose of charting your risk history is so that you can review and examine those risks, and analyze their outcome. How many of your risks had positive outcomes? Do you remember how you felt and what elements caused you to go ahead and take the leap? Can you apply those factors to a current risk that you're considering? Chances are that you'll find that many had positive outcomes and that you feel slightly exhilarated after thinking about them. In addition, you might feel more comfortable and certain about the risk you're considering.

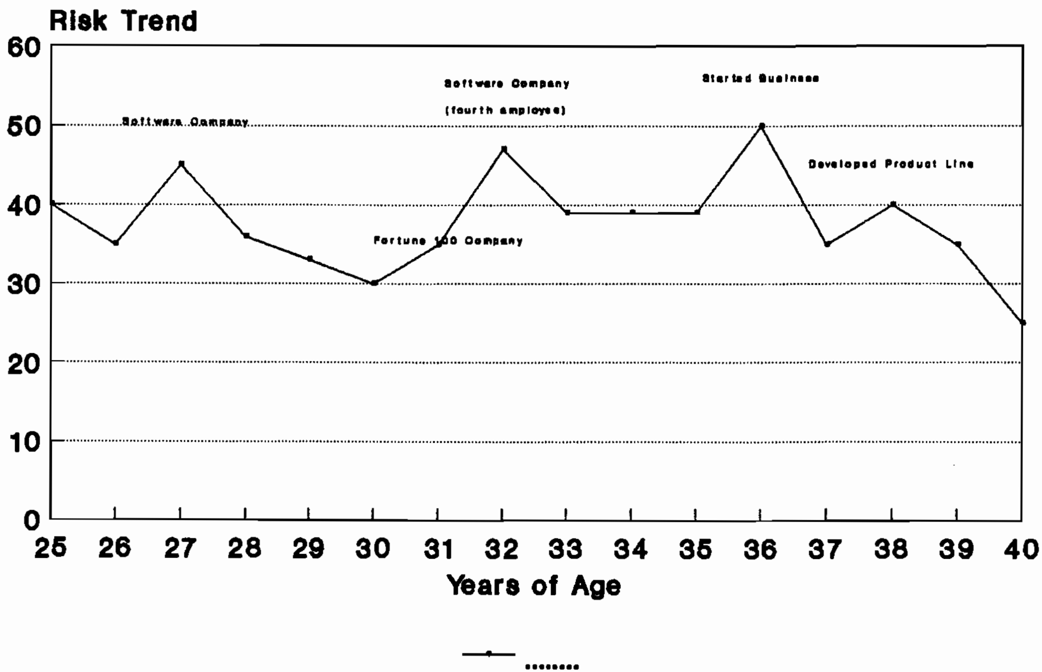
Another way in which you can encourage yourself to take risks is to carefully calculate the possible outcomes. This means more than casually considering what might happen. It means that you must determine the positives and negatives that might result in your taking the risk. For example, if you're considering a change in careers, first, list the factors that caused you to consider the risk in the first place. This might

FIGURE I

Charting Risk History

2809-6

FIGURE II
Career Risk Chart



include things like additional income, job satisfaction, peer admiration, personal growth, and personal pride. Then, list the possible negative outcomes. What if you wind up making less money instead of more? How little is too little? What if you wind up feeling a lack of confidence because you are unfamiliar with your new role? How will you deal with that?

Listing the positives and negatives does two things for you. It allows you to evaluate whether or not you can deal with the worst possible outcomes, but more importantly, it allows you to anticipate some of them, and put strategies in place for dealing with them ahead of time. Finally, it allows you to measure the reward versus the risk, and thereby decide whether or not its worth taking the step.

Another important aspect becoming comfortable with risk, is practicing whatever it is your considering doing. Let me give you an example. About ten years into his career, a successful Systems Engineer considered a career switch to software sales. He had worked with customers extensively, so he was comfortable with dealing with their problems and concerns, but he wasn't at all comfortable with two aspects of sales; telephone prospecting and asking for an order. When he mentioned his fear to one of the Sales Reps he was working with, the Sales Rep suggested that he do some telephone prospecting before taking the job. The Sales Rep gave him a list of prospective customers in his territory who were the least likely to be interested in the company's software product. After coaching the Systems Engineer, the Sales Rep encouraged him to try any approach he liked that was comfortable for him.

"I don't think I'll sell anything to these companies anyway, so don't worry about losing a sale. And who knows. Maybe you'll interest them."

By taking the pressure of "success" off of the Systems Engineer, the Sales Rep gave him a non-threatening environment in which to practice his skills and determine if this was the right job for him. What was the worst thing that could happen? The Systems Engineer wouldn't lose his job. He couldn't even lose a sale since the Sales Rep had already written the companies off. The key here is that the risk was removed for a short period of time, allowing the Systems Engineer to make a more objective decision.

If you're getting ready to go into the boss and ask for a raise or a corner office with a window, practice first with your spouse or a friend. Be sure to give them a profile of your boss first, so they can simulate his or her responses. Encourage them to make it tough on you. Then you'll be ready for anything.

Before taking the risk you're considering, it could benefit you to seek out and spend some time talking to other people in your life who have taken similar risks. This might be a family member, a friend, or a co-worker. Ask them what steps they went through in their decision making process. Talk to them about why they think they were successful or unsuccessful. It always helps to know that others have walked in your shoes.

Finally, one of the most important aspects of risk taking is your belief in yourself. The competitive diver who allows herself to feel a momentary lack of self-confidence as she takes that last bounce on the diving board will not execute a perfect dive, and might even injure herself. She's got to go out to edge of the board thinking that she's a championship diver; better than any other diver competing. That's how you must approach risk, once you make your decision to take the plunge. Tell yourself that your an intelligent capable person. Remind yourself of your past successes. Before you take the risk, imagine yourself in the position of success. Imagine walking into the boss's office and approaching him or her with your request. Imagine that boss offering objections, and imagine yourself meeting those objections. Finally, imagine the boss agreeing with your point of view. Go through this process several times. You'll find that you'll be much more comfortable when you actually make your move. Imagine a scenario enough times, and it can seem almost as if its happened, giving you the feeling that you already have experience with the risk you're taking.

If things go sour, remind yourself of your past failures, and how you were able to successfully recover from them. You're still around and even though you may have had some setbacks, you've learned and grown through trying new things.

If you still find that you're not ready to take a risk that some part of you wants to, look at the reasons why you're avoiding the risk. There are a number of possibilities. You might have a realistic fear of the possible negative outcomes. If so, you might want to put the thought of the risk aside for awhile and re-examine it after some time has passed.

You might lack the self-confidence required to take the risk. See if you can find some formal training that can help you to feel more confident. Depending on the area you live in, you can often find courses on assertiveness, self-confidence, or, more specifically, courses that will give you the skills you feel you need before you take the risk.

If you're avoiding a risk due to financial or other responsibilities that you have to your family or loved ones, the time might not be right for the risk. Determine what it is that you can do that will put you in a position to meet those responsibilities while you take the risk. Perhaps you'll need to delay for a year or two while you set aside some money for the possible down side to your risk. Maybe you need to wait until your children are older. In any case, don't turn away from the risk, but rather, put together a plan that will allow you to do what you want to do.

If you've decided to forego the risk because the possible negatives outweigh the possible positives, you've probably made a good decision; as long as you've actually gone through the process of evaluating the risk in depth.
If, on the other hand, you're avoiding the risk because you're comfortable with where you are, take a second look. Most of us consider risks because we have a basic belief that we can improve our lot in life, and because we believe that to truly live you must risk. Don't pass up what might be an exhilarating experience, and what at worst will be a learning experience, because you want to avoid the unknown. Whatever your age, you're too young to avoid risks at the expense of growth.

DESIGNING FOR CHANGE

A Workshop

DANI WEINBERG

WEINBERG & WEINBERG
Rural Route 2, Box 32
Lincoln, Nebraska 68520
402/781-2542

DESIGNING FOR CHANGE
A Workshop

Change is an emotional, chaotic, and subjective life process. We cannot successfully introduce new technologies without taking this fact into account.

In this workshop, we examine

- the human experience of technology transfer
- a powerful model for understanding the change process
- appropriate interventions for each stage of the process

The workshop moves back and forth between theory and practice, using participants' own experiences of change as material against which to develop and test theory. Participants work on their own specific organizational cases, learning to analyze the situation and designing strategies for managing change.

THE GREATEST COMPUTER SECURITY THREAT

by Vladimir Volokh
VESOFT
1135 S. Beverly Dr.
Los Angeles, CA 90035 USA

The problem of computer security was definitely not invented by software vendors -- just read the newspapers every day.

Computer crimes come in different flavors:

- simply reading sensitive data (prices, customer lists, etc.)
- modification of data (payroll rates, shipping information)
- sabotage (viruses, time bombs, intentional system crashes)
- software theft
- unauthorized computer use
- defense-related crimes
- and more...

Security-minded authors have written many books, as well as articles in HP-related publications, on this subject; we at VESOFT became involved in the HP 3000 security industry in its very infancy, presenting computer security papers at HP conferences in Berlin (1981), Copenhagen (1982), Anaheim (1984), and, most recently, at the INTEREX security seminar in 1989.

And yet not every HP3000 computer is secure! The word SECURITY is misleadingly simple, simple enough to make many people think that they have adequate system security without fully thinking out what HP 3000 security really entails.

The issue of computer security is actually very complex -- it involves:

- physical security (guards, dogs, locks)
- system set-up (accounts, groups, users, capabilities, access, etc.)
- logon security
- remote access security (dialups, lans, etc.)
- batch access
- logoff security (can people just walk up to an unattended terminal and use it?)
- IMAGE security (have you EVER changed your database password?)
- file system security (why does MPE have a :RELEASE command?)
- application security (who is allowed to print checks?)

- data encryption (fields, files)
- back-up and disaster recovery
- and more... much more...

System security is every bit as much a primary concern of any DP department as the actual applications running on the machine.

SYSTEM SET-UP

Look at your accounting structure first:

- What accounts do you have (check it by using :REPORT X.@)?
- What groups (use :REPORT @.@)?
- What users (:LISTUSER @.@)?
- Which capabilities do each of them have (SM, OP, PM)?
- What kind of access (Read, Write... -- for ANY, AC, GU...)?
- Are all of these entities passworded, or only some of them?
- Are some of the existing passwords too short or too obvious?
- How often are they changed (if at all)?
- How many various levels of UDCs are set on your system?
- And if you rely on them, how easy is it to bypass them?

LOGON SECURITY

Logon to the HP seems to be quite secure with ACCOUNT, GROUP and USER passwords. Or is it? Look carefully:

- MPE error messages at logon time are too *"friendly"*
- Passwords are **readable** combinations of up to 8 ASCII characters
- They are either easy to guess or difficult to remember -- and users write them down (sometimes even stick them to the terminal)
- They are often shared or simply disclosed
- Seldom changed
- If users use the session name (:HELLO MARY,MGR.PAYROLL), it only looks better -- the session name isn't enforceable and the password is assigned to user (MGR) anyway
- Yes, the MPE password is **assigned**, so account manager is the first suspect (and all SM users too)
- Is it easy to enforce shifts (time restrictions on logon)?

- Can payroll be run in the computer room (from LDEV 20)?
- Or can it be done on the weekend?
- Can end-users ever see ":"? What can they do then?
- What is better: to forbid most MPE commands via *"clever"* UDCs or to let users execute only some commands and subsystems?
- And if you have a logon UDC which brings them into an application, how about some other applications (e.g. HPMAIL), some utilities? Should users constantly change their logon ID?

REMOTE ACCESS

Remote access to the computer is common nowdays: dial-up, DS, NS...

- Who knows your dial-up telephone number? Your former employees, current employees, telephone company workers, HP SEs...
- Simple question: what to do if a person leaves the company? (Change all passwords on the system, unplug dial-up forever, request to change your dial-up number...)
- We have a horror story to tell you: one of our customers did change their dial-up number, but... the telephone company set call-forwarding onto it (you know the message -- *"The number has been changed, the new number is..."*)
- And if there are two or more computers linked together, can any programmer access the production HP/960 from the development HP/42?

LOGOFF SECURITY

Logoff is also a problem: you should realize that unattended sessions constitute a major threat to system security. The more sessions you have (it can be hundreds on XL) the less control you have.

- Remember that an unattended terminal is a convenient way for some people to use your system without logging-on.
- Also, if the session is left on after hours and keeps some files open these files might not be backed up.

BATCH SECURITY

Batch security is as important as on-line, but...

- MPE requires passwords to be included in job cards, so a typical job card looks like this :JOB FULLDUMP,MANAGER/SECRET.SYS/QWERTY
- This makes passwords easy to read by unauthorized people and difficult to change on a regular basis by people responsible for system security (it might be you)
- There are some other important things built into streams -- lockwords, database and/or application passwords, etc.

- The situation is somewhat better if all of your streams are in groups with X:ANY,R:GU access -- but try to verify this

IMAGE SECURITY

IMAGE security had better be good -- that's where our most important data usually is. However

- Passwords (up to 63 of them) create the appearance of good protection of the base, sets, and entries

- But... these passwords are often built into sources -- intrinsic DBOPEN requires this; source code is compiled and guess what? IMAGE passwords are **never** changed! The situation is so bad and continues for so long, that HP users seldom recognize this kind of danger.

- It's even worse when using some application packages -- all customers of this package have the same password. Would you buy a car with the same key for everybody else who buys the same car?

- Some system managers sense something wrong in this area and set a lockword on QUERY. It's better than nothing, but what about other database retrieval tools or custom written programs?

FILE SYSTEM SECURITY

File system security in general is very important. A couple of questions come to mind:

- How many files on your HP are released?

- Even worse -- are these files in PM groups?

- And how do you :SECURE hundreds of them? Are you the *"creator"*?

- Which files were accessed on your HP over the weekend?

- How many programs, and which ones, have PM capability?

- Is it possible to :FCOPY the object code of your programs in ;CHAR mode and see all the 'built-ins'?

- Do you like the recent ACD (Access Control Definition) enhancement for MPE/V file system, which, in short, links particular users to the file?

- If so, before using ACDs, think about selection of these files later -- they will be as invisible as :RELEASEd files; think also about setting ACDs for groups of files, saving ACDs after editing text files, etc.

Having said all of the above let's ask ourselves:

What is **the biggest** computer security threat?

And it seems that the problem lies in the wrong approach to risk management on the part of DP personnel. As long as system managers continue to count on users' ignorance, on end-users being *"good"*, on having **only one** dial-up line (yes, we've heard this one too) and such, company assets -- and some people's resumes -- will be in danger.

Paper #3005

Turnkey Solution Projects

M.N.Chaturvedi

Datamation Systems
P.O.Box: 7310
Abu Dhabi, U.A.E
+971-2-727767

Abstract

Successful approach of a large Turnkey Computerization projects, from Concept of Business Systems Studies to Implementation Stages with key elements of software and manpower are being discussed. Definition of work packaging of manageable size and its formulation are being discussed with examples. Right profile of Resources and its Management with the complexity of human nature has been identified. Structured communication and in-between lines communication, during the Project Cycle, is being stressed for their strength and weaknesses as well as applicability on the situation analysis. Techniques of Project Management and Probability Analysis and their impact on Software Project is elaborated at great length. Strength of resourcing of multidisciplinary cultural streams, a growing necessity for the future collective skills, are being discussed. Dimension such Creativity, thoughtfulness and Managing the Inner Conflicts, are discussed with their importance in the Project life cycle. As conclusion the key elements such as professional manpower skills, equipments, cost effectiveness are organised. To sum up, a Lead Handling of successful Turnkey Project and methodology is being evolved as an example to follow up.

Introduction

Evolution of solution is continuous process. The solutions keep on varying on the basis of requirements alongwith other factors. The major business system developed for the same functionality some years back could be different in approach and objective in the current time frame. The solution developed is function of many factors. Few of them are environment, human behavior, technologies available and needs of the system. A success of the project shall be dependent on successful handling of these factors in combination to achieve the objectives.

Business Systems Study

The main objective of business systems study is to define the boundaries and scope of work and document the needs of the system. The requirements of the systems are evolved through structured dialogue with the Users of the system and with the managers of the system. Various techniques, as available, are useful such Top down, and bottom up method. Single methodology may not be useful to the desired result however combination of both the approach coupled with the heuristic approach could be best fit in some of the intricate systems.

The key factors of the project are definition of requirements. The requirements are the basic needs to be fulfilled by the system, however it is useful technique to extract the requirements which are key but generally latent or eclipsed during the discussions. These factors are important and play vital role in the successful implementation of the system.

Business system's study must identify the objectives of the system, such as gains of the new system, accountability, information needs and its flows, the tools/equipment for fulfillments, cost implications, acceptance criteria and not the least "What is not covered" which generally is assumed to be covered by the reader of study report. It is imperative to asses the cost effectiveness of the solution. In general fitness of package and its evaluation plays the key important role for the solution and success of project. At the Business Study level, there are two sets of requirements to be developed

- Necessary package characteristics, which are usually at higher level defining the direction and mode of operation. These are usually few requirements of such nature. These are the requirements that are used for first pass screening of products and that will cut down the horizon for detailed product review.

- Desirable package characteristics, defining all the detailed systems requirements that could be thought of but may not be absolutely necessary. These can be used for detailed product evaluations. These are best grouped by category e.g

  * Functional requirements
  * Architecture or hardware requirement
  * Database considerations
  * Project management considerations

The statement of requirements should include "must not" and undesirable characteristics. Specific checklist is required to be developed for consistent and objective evaluation process.

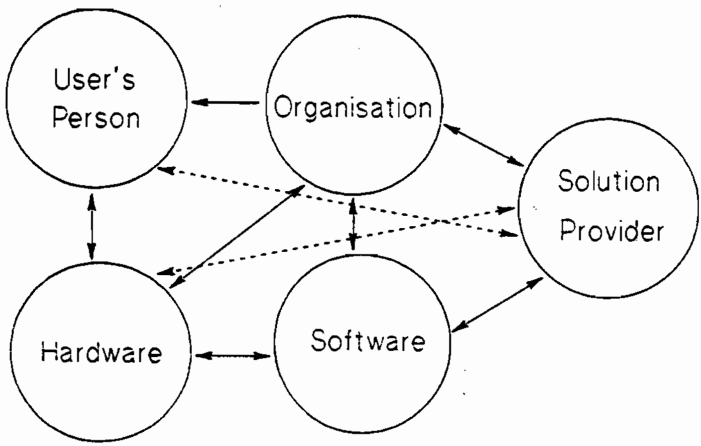
Concept Design:

Conceptual design conceived after systematic study is the framework of system design. It is equivalent to defining direction of effort and its impact on meeting the requirement and which is extensive and voluminous. It is similar to the situation of providing direction, and if direction is lost the objective becomes unachievable. Concept design is successfully handled by structured techniques as well artful techniques well groomed through experience. Generally the CASE tools are the basic tools which are used for the iterative process of concept design. The criteria for closing the design are not easily verified. It is complex situation to check and meet the conversion process of concept design. Close looping of information flow is one of the basic and initial check procedure. It is also possible to check the Macro level design through heuristic approach.

Right selection of CASE tool and technique for concept design will result in reducing or completely eliminating the rework process during test phase or implementation phase. The key essence lies in completely eliminating the rework for a successful project. Failure of project is mostly attributable to the rework of the project.

Concept design shall be revolving around basic entities to fulfill the needs and their interactions effectively.

User's persons, organization entities, software to meet the functional needs as well hardware tools and equipment are the interactive view to solution provider and looked from proximity on equal weightage factor or priority. There have been situation where concept design must meet the need of the desire factor of whole environmental system for success at the end of the day.



Project organization

Task Force Organization

Project shall be manned by a team of System Analyst, Programmers, depending upon the requirement on Task Force Concept.

The following project organization is formulated, maintained during project cycle as key success essence of project execution philosophy:

System Analyst and Programmers are allocated on the basis of requirements of each application, target date, commitment and qualitative need of the applications.

One of the main factor of project execution is project review team meeting, which is suggested to be held every week/fortnight and reviews the following aspects in short and to the point:

- Review work done during current period
- Work assigned for next period
- Reallocation of manpower
- Technical Difficulty, if any, on any area of project
- Methods improvement on critical resource consumption area
- Any other outstanding point.

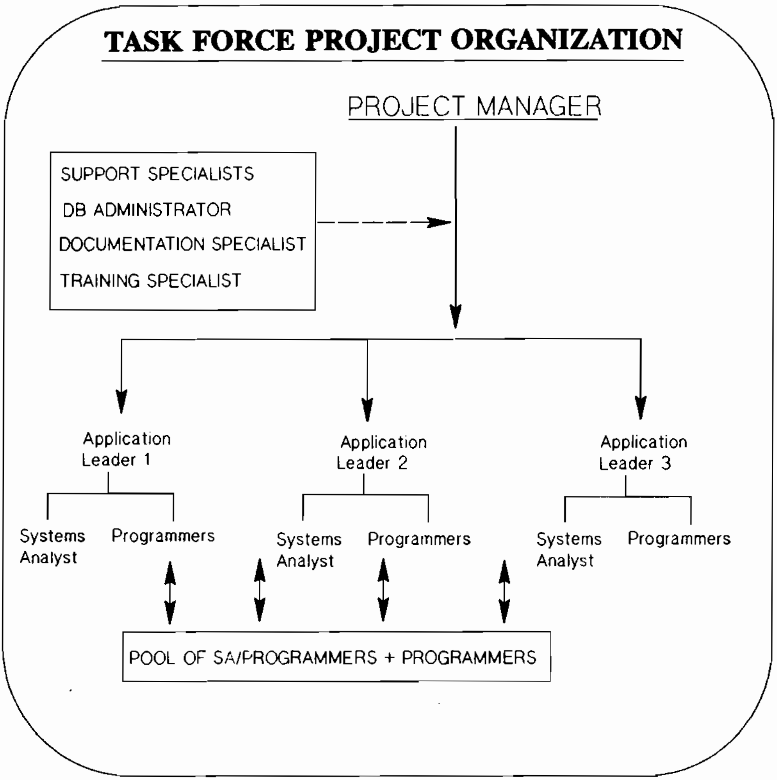
Above are brief techniques used in for execution and control of projects.

Involvement of Team Members:

As shown in the chart, the application shall be handled, planned, developed and monitored by one Application Leader who will be reporting to the Project Manager. The application Leader shall be meeting the Development Team on as and when required basis, but the meeting must be once a week/fortnight to resolve, smoothen the interfacing activity/program/file design and testing procedures during the development process.

Once the conceptual design and the requirement is discussed out by Project Manager for onward detail design of the Application, the Application Leader shall prepare requirement specs, concept design, detail design, program design and all the test conditions. Project Manager and Customer's authorized representative shall be approving the detailed design.

On approval, Application Leader shall be assigning Programmers to accomplish the job. Programming job could start to crash the project duration suitable number of professional programmers are put for specific time to achieve the target date.


TASK FORCE PROJECT ORGANIZATION
PROJECT MANAGER
SUPPORT SPECIALISTS
DB ADMINISTRATOR
DOCUMENTATION SPECIALIST
TRAINING SPECIALIST
Application Leader 1
Application Leader 2
Application Leader 3
Systems Analyst
Programmers
Systems Analyst
Programmers
Systems Analyst
Programmers
POOL OF SA/PROGRAMMERS + PROGRAMMERS

Resources required from user organization

It is envisaged to have the following resources available from user organization during project execution.

- Project Manager/Coordinator

  On signing the agreement, it shall be imperative for the user organization to assign a Project Manager/Coordinator for Successful conceptualization, development and implementation of computerization project. From time to time, he may get assistance of Application Expert within his organization. Project Manager shall be the single point of contact for all the activities from all the user's side.

- Availability of staff for requirement study:

  As and when required, user's shall provide all the facilities, appointment with the existing staff for Application Specification, Testing and Approval phase.

- Support during conversion:

  During the existing Data File Conversion, facility of the existing system etc. along with clarification of system is expected to be given to supplier appropriately.

- Key punching:

  During parallel months processing, it shall be essential to Key Punch data very effective in line with the schedule.

- Other facilities:

  For training of users on application, class-room facilities with possible demonstration terminal shall be required.

Motivating the Team

Every team player has strength of his own which is vital to be used at right place at right time of the project cycle. Human skills of management are necessary in the project manager to arouse the creativity strength of the members which gives the collective contribution to the project. Creativity is the base of solution project and there is no substitute of creativity in these projects. It has been observed that challenging job opportunities and job enrichment provide the most important attribution to the motivation. Professional skills, which needs to be nurtured for result of high achievement, are motivated through the stream of management support. Vertical loading of professionals shall be most appropriate to motivate and get the higher productivity. Principles of vertical loading is provided in the Chart given here. (Reference is acknowledged to Frederic Herzberg for HBR article "How do you motivate employees").

Further, motivation of professional team members must be done with great care in order to avoid the negative impact of the personality factor as well inner conflicts. These negative factors, if observed in the team, must be handled with positive attitude of the project manager without loosing the time. The project manager having training in behavioral sciences could be rightly handle the team for the positive results.

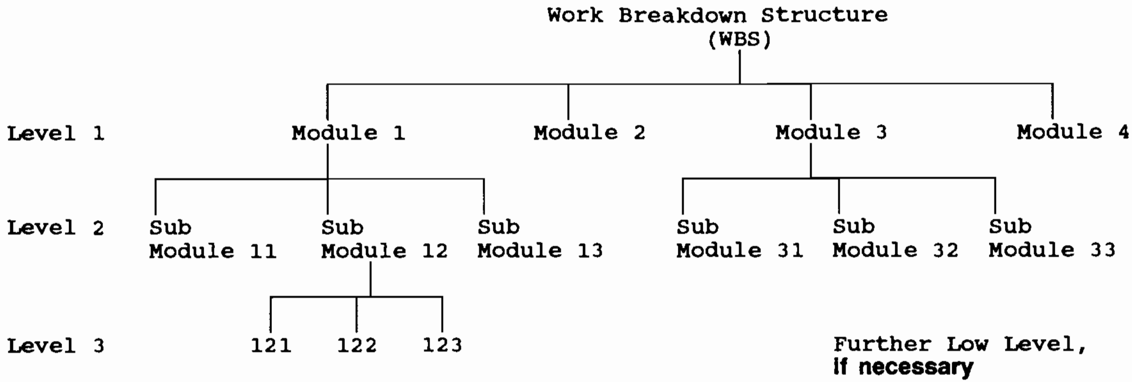
Job Motivation

Activities	Motivators involved
1. Increase the accountability of individuals for own work	Responsibility, recognition
2. Assigning the complete logical work unit	Responsibility achievements & recognition.
3. Assigning additional authority in own activities, job freedom	- do -
4. New areas of Technical excellence, difficult untouched areas	Growth & Learning
5. Specific areas, specialised task leading achieve their expertise	Responsibility, Growth & advancement
6. Sharing of contribution on resolving complex problems or bottlenecks	Recognition
7. Freedom of work keeping accountability, within the framework	Personal achievement & recognition.

Work Breakdown Structure (WBS)

In a solution project, we need to breakdown the work packets of manageable size from point of view as

- Functionality
- Independence
- Communication
- Normalized efforts



Examples of Project: Financial Systems

Module: - **General Ledger**
(Level I) - **Accounts Payable**
Etc.

Sub Module: (Level II

Option I
- **Masters**
- **Transactions**
- **Reports, etc**

Option II
Sub module on functional
- **Invoice Processing**
- **Payment Prepration**
- **Cheque Printing etc**
- **Interface**

etc.

These levels are also useful for screen flow diagrams.

Project Progress Measurement

WBS Estimation Hours/Weight Matrix

(Estimates are in manhours)

WBS Code	WBS Description	Mile Stone						Total hours	WBS Level Weight
		Requir. Specs	Concept Design	Detailed Design	Develope Code & Unit Test	System Test	Imple-mentation		
Mile Stone Weight		.15	.15	.20	.20	.15	.15		
1.	Level I AP								
	Level II								
11	- Invoice Processing	50	75	100	200	50	75	550	0.48
12	- Payment Prepration	100	100	100	150	50	100	600	0.52
1.	AP	150	175	200	350	100	175	1150	0.45
2.	GL	200	250	250	400	150	150	1400	0.55
	Project Total	350	425	450	750	250	325	2550	

Note : Figures represent example for illustration purpose only

The purpose of WBS shall be to help in organization of project and its management in order to increase the effectiveness, quality and reduce total cost of the project. WBS is analogous to organisation of any company to fulfill the objective. As organization of the company is important for smooth operation and function, similarly the project needs WBS for the structured and smooth execution. WBS is based on top down structure of project and logical breakup of project into many manageable levels.

One of the example is prepared in the chart of WBS is a simple example for illustration. The estimation of effort is done for each lowest level of the WBS. The lowest level is unit which could be completed and flows through the milestones of project cycle. The weights of each milestone are assigned from the overall efforts and its importance on the basis of experience.

The weight factor of each lowest level of WBS is assessed quantitively (proportional to quantum of work) and worked at the higher level of WBS through the relative efforts. These weights for WBS packets are index of efforts on normalised scale. The efforts in the matrix are given in the man hours.

- Progress of submodule is

  estimated as = $\dfrac{\text{Sum(Milestone completed hours X Weight of milestone)}}{\text{Total submodule hours}}$

- Progress of Module

  = Sum(percentage progress of submodule X Weight of submodule)

- Progress of project

  = Sum(Percentage progress of Module X Weight of Module)

Further performance of project can be worked out on the basis of actual hours spent. Index of actual hours spent for the all the completed milestone to estimated hours of completed milestone shall indicate the performance.

Detailed Design

Detailed design is the activity performed by the analyst on the basis of explosion technique related to concept design. Basically the concept design which is base integrating the specification and reflecting scope of work and boundaries is exploded into finer details of detailed system design. Main design consideration may vary from third GL approach to 4GL approach. In both the approach the coverage to logical and physical data organization, its access method are the key essence. Techniques using the CASE tools provide the structured through process as well clarity of solution. Couple of design alternatives are required before reaching the near global optimum solution. It is utmost important to design more than two solutions for the same objective in order to ascertain the better method. It shall be complex modeling to quantify the variable and reach the global optimum solution through multiple iterative approach of CASE tools. However, through heuristic approach and on basis of experience it could be possible to reach the near optimum design solution.

The factors which shall be guiding the assessment of application design could be

- Functionality
- Storage of Data
- Accessibility
- Maintainability
- Simplicity

Once the flow of data and detailing is done the design shall revolve around the data dictionary definitions and ensuring the uniqueness of field definitions. This shall be further help in eliminating the redundancy. We are talking about the global data definitions rather than module data definitions. As the business systems are modules to modules defined, designed and integrated a high degree of interfaces are there which plays important role in effectiveness and success of the software project. In general the solution design needs to be modular for functionality, design development and implementation depending upon the logical work breakdown structure, discussed in the paper, without loosing the significance of integration of modules. Software Engg. tools and techniques of CASE tools are most appropriate for design and validating the interfaces.

These techniques shall help in reducing the reworking of the project during implementation stage. In practice, Reworking and backward change and modification is one of the major factor attributing negatively to the project and must be controlled by project manager, with vision, as well every member of the project team.

Development

Real production process starts at development stage. This activity is similar to production workshop where units can be produced by putting the large resource bank and this activity becomes vital from the point of view of crashing the project schedules. It can offer maximum amount of flexibility for crashing the project. Technical skills, availability of coding tools, productivity aids and not the least motivation plays the key role in productivity during this stage. Independent units are defined for execution development with their estimated efforts, target date, conditions of acceptance etc. The key factors for increased productivity is to automate the process of development. This could be achieved by exploring the common factors used in module/code development. It has been observed that development effort and thereby time reduces in geometric progression with the commonality factor/utilities. It has been further observed that in few cases 3GL approach with high degree of commonality factor has taken much less time and effort compared to 4GL without common factor considerations, in the same environment of machine.

Software development workstations generating the code through the software engineering CASE tool is the highest in productivity through mechanization. The factors which shall affect the software development are

- Provide highly interactive responsive and dedicated environment
- Automate many software development tasks
- Provide pictorial view of software by means of graphics, such as logical flow diagram, structure block flow of the programs etc
- Prototyping in early stage
- Automate code generation
- Automate the checking validation process
- Plan to reuse the code in modular approach method

Testing

Testing of the units, modules and complete integration test for system is performed through combination of skills such as application knowledge, machine environment, languages used in the development of system. It is imperative to use the testing standards methodology and structure for testing procedure. As the higher degree of testing function improves the quality and reduces the rework at later date during implementation stage, this process is packed with the thoroughly normal as well abnormal conditions at the same time. It is always easier to perform the testing of batched system rather on-line system where human to screen functions are repeated. The test plan should include

- Boundaries of Test
- Normal conditions
- Abnormal conditions
- Convergence criteria
- Time frame
- Responsibilities

Definitions of boundaries are important from the point of view of limiting the non productive situations as well coverage of test horizon. These are required to be well defined for the team from project managers point of view. Further, these boundaries guide the level of testing procedure such as unit test are necessarily to be performed with in the unit test data and interfacing needs may not be necessary. Where as integration test must be across the boundaries of module. Distinct process of capabilities are required in two zones of testing procedure.

Mostly normal conditions are put through the testing phase. It is necessary to bleed the live system to extract the data for the testing plan. However the analyst is required to supplement the test data with abnormal conditions derived out of his application knowledge and analytical ability. To close the loop these abnormal conditions must have convergence criteria which are normally defined in the test plan to conclude the testing process.

One of the key feature of testing procedure mostly related to module test and higher level system integration test is to continue the testing procedure on modules even if slight deviations are observed in the previous step. In general, attempts are made at local level with a short vision during the testing procedure to rectify the error immediately and proceed. This may lead to continuous patching and rework of the system during system test runs.

It could be good practice to limit the accesibility of source programs from the machine during test procedure to observe the behavior of test runs and document the results. These results must be reviewed by whole project team to conclude the deviations, root causes and plan of action in the total system approach. Peace meal corrective procedures may lead to time over runs, cost over runs in totality of the project as well delivering substandard quality system. Couple of test runs as iterative procedures as above shall be completing the testing milestone.

Test data may be required to be preserved for repetitive test situation in order to reduce the wastage of efforts and time on this procedures.

Implementation

Implementation of large software project is complex and intrinsically difficult task. Nearly most of the software project face numerous difficulty during implementation since it is function of organization structure, user interface of the application and dovetailing the before screen function with after screen logistics. The key factors affecting the success of implementation are:

- Top Management support
- Organization structure, its size
- Geographical locations
- Application knowledge
- Human inter relationship skills
- Environment knowledge

Larger the organization to implement the system its problem are different in nature and complex. In general, efforts to implement the software project are proportional to geometic progression of organization size. A well structured organization shall be offering most streamline operation of the implementation process, since the information needs and its flow could be simulated during requirement analysis through software engineering tools. Any redundancy in information flow in the organization which may not be offered through structurally developed system, shall be facing the resistance to change. If these software change forces evolved during the implementation phases are not controlled through positive approach, it shall amount to changing the structured system application to suit redundant information needs of the organization. In order to help the process of implementation, the organization and methods study and its implementation could be recommended for the smooth acceptance of the system.

Any software project could be successful only if the Top management support is provided. The new system implementation shall get the resistance from the previous one due to its familiarity of use, etc however the new system should be offering the more benefits to the users and management, and value attached to these benefits should be presented to overcome the resistance to change.

A continuous reporting and feedback system designed to interface the Top management shall serve the function of catalyst as well boost up process to the whole implementation process. It is necessary to get the management approvals and directives to the organization for solving hold ups and bottlenecks during the implementation process.

The user's reactions must be analysed in light of benefit and can never be ignored. Some comments from the users are so important at the time of implementation phase that these lead to the next release of software with high degree of effectiveness. That is how the contribution to software enhancements are evolved.

Training to users is vital to the success of implementation. Without proper training user's appreciation is limited and may result in software portion unusable and providing the poor effectiveness of the software to the organization hence limited benefits. Most of the resistances for change are overcome by training given to users in the start of the system. At the end of the training, which is structured class room discussion with extended workshops, the evaluation could be useful to provide the feedback of the system. Project is released to organization after the successful implementation process.

PAPER NO.: 3007

TITLE: High Water and Broken Bridges

AUTHOR: Gaylord Maines
Bahlsen, Inc.
One Quality Lane
Cary, NC 27512
919-677-3260

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO.: 3009

TITLE: It's Tough When There's No Can Opener

AUTHOR: Mark Symonds
IISI
63 Nahatan Street
Norwood, MA 02062
617-769-7511

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO.: 3011

TITLE: The Monday Morning Blues

AUTHOR: Kathy Spanel
GBS Consultants, Inc.
6087 S. Quebec Street #101
Englewood, CO 80111
303-721-0770

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO.: 3012

TITLE: Managing a Power-House Shop

AUTHOR: David G. Robinson
PowerSpec International
403 Cross Lake Drive
Fuquay Varina, NC 27526
919-552-8049

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

3505
INFORMATION MANAGEMENT TECHNOLOGIES IN THE 90'S

Orland J. Larson
Hewlett-Packard Company
19091 Pruneridge Avenue
Cupertino, California 95014
(408)447-1197

ABSTRACT

Enterprises have come to depend more and more on the accessibility, accuracy and timeliness of information. During the next few years, new database technologies will provide solutions to many of the problems and difficulties facing today's MIS and user departments.

This paper begins by reviewing the changing world of information management and the challenges facing MIS. This will be followed by the major trends associated with information management over the next several years. These include the continued acceptance and importance of relational database technology, the increased interest in distributed database applications and the emergence of the new object- oriented database management capabilities. Finally, this paper will address the significance of these technologies in the cooperative computing environment.

INTRODUCTION

THE CHANGING ROLE OF DATA PROCESSING

Before looking forward into the 1990's, we should first look back at how data processing has evolved over the past twenty years.

The general trend of the 1970's was the use of centralized computers and resulted in systems that were often difficult to use, inflexible and usually did not meet the end user's needs. Database management systems (mostly hierarchical and network) became widely used and provided the basis for on-line, interactive applications. In addition, the computers and operating systems provided programmers the capability of developing applications on-line, while sitting at a terminal and interactively developing, compiling and testing these applications. The end users were also provided easy-to-use, on-line inquiry facilities to allow them to access and report on data residing in their databases.

During the 1980's, the emphasis was on the decentralization of data processing. This includes the proliferation of personal computers which has resulted in both the "islands of automation" and the corresponding "islands of information" problems. This in turn resulted in reduced control of corporate data for the MIS department. In addition, relational databases became commercially viable and experienced wide acceptance even though performance was often an issue. Relational database performance has now improved significantly; and they are currently proving effective in on-line transaction processing (OLTP) environments. Software tools such as 4th generation languages (4GL's) continue to be used successfully as an effective way of developing applications through the concept of information systems prototyping. This required that the end user be more involved in the development of systems and has resulted in more effective systems that meet the users' needs. This has helped to reduced the backlog of applications but usually also has contributed to the "islands of automation" problem.

As we move into the 1990's, relational database will continue to gain wider acceptance. It is the enabling technology and the basis for distributed database management, which provides transparent access to data which is distributed over several sites.

There are also new technologies called object oriented database (OODB) and object oriented programming systems (OOPS), which will manage more complex data structures and will result in improved programmer productivity and more flexible systems.

An additional technology, cooperative processing, is evolving which will help integrate those "islands of automation" back together and allow for the data and programs to be accessed and shared in a cooperative computing environment.

THE IMPORTANCE OF SQL

According to a recent Gartner Group Report, in 1988 only about 7% of the applications developed used relational database or Structured Query Language (SQL). However by 1992 their prediction is that 65% of applications developed will use SQL.

There is no doubt that SQL will be the basis for applications developed in the 1990's. One of the main advantages of SQL is data independence or the immunity of applications to changes in storage structure and access strategy. Another main advantage of SQL is the simplicity of the underlying relational model which is the easiest to understand - at least at the most basic level. In this model, data are represented as tables, with each horizontal row representing a logical record and each vertical column representing one of the attributes, or fields, of the record.

The following are the key points associated with relational technology:

* Relational concepts are easy to understand and use.
* SQL is a multifunctional language
  - Database definition and creation
  - Data retrieval and manipulation
  - Authorization and security
  - Transaction management and recovery
  - Database environment management and restructuring
  - Interactive and programmatic use
* SQL allows you to specify which information you want - not how to retrieve it.

* SQL increases programmer productivity and raises programming closer to the level of problem solving.
* Data independence is ensured and minimizes maintenance of programs
* Data access can be automatically optimized as the database structure changes.
* The DBA has unprecedented power and control over the database.
* New systems can be implemented much faster.
* SQL assists in cross-system connectivity.
* Relational databases provide a cost effective, powerful solution.
* Basis for a true distributed database environment.

There are, however, some areas of SQL that need improvement. The current SQL standard is missing some important features and some of the standard features are implementor defined. In other words, no vendor fully supports the complete "standard" and no two SQL implementations are exactly alike. These inconsistencies will lessen as new levels of standards evolve. Organizations like the SQL Access Group are also working hard to resolve these problems.

As we enter the 1990's, relational database is becoming the dominant technology in today's information management marketplace. There are several enhancements planned to improve functionality and performance. It eventually will be appropriate for most applications and gain wide acceptance by all users.

Relational databases can improve the quality, control and accessibility to your organization's extremely important and valuable information resources. It can result in an improved competitive position by aiding business analysis that can help to determine ways to improve products and services.

Unlike non-relational database environments, relational databases adapt easily to dynamic business requirements. In addition, unrestricted access to important data means better information for more effective decision making.

Relational database can also have a positive effect on many MIS development environments by reducing the application backlog and reducing the time and cost required to develop applications. The improved database flexibility and ease of change can also result in a significant reduction in the maintenance of applications.

Overall, the use of relational technology can increase the MIS professional's effectiveness and productivity, which results in improved user satisfaction and confidence. Choosing relational now will position your organization to take full advantage of the technological advances of the 1990'S.

DISTRIBUTED DATABASE

One of the hottest topics in the commercial database world is the growing trend towards the use of distributed database management systems. After many years of research, distributed databases are becoming more viable. However, there is still much to be done to provide more than just read access to distributed data. Chris Date, one of the world's leading experts on relational database, recently presented a paper (see reference 3) in which he provided a working definition of distributed database. "A distributed database system is a system involving multiple sites connected together in a communication network, in which each site is a database system in its own right, plus a user at any site can access any data in the network exactly as if the data were all stored at the user's own site. Thus, a DDB is a virtual DB whose components are physically stored in a number of distinct real databases at a number of distinct sites."

Chris Date follows this working definition with an "alternate" or more elaborate definition. "A distributed database system is a system that allows an arbitrary collection of relations, from an arbitrary collection of databases, on a variety of different machines, running a variety of different operating systems, connected by a variety of different communication networks to function as if they were all stored in a single database on a single machine. The user is completely insulated from all details of distribution."

Distributed databases can allow the structure of the database to mirror the structure of the company, while simultaneously solving the "islands of information" problem. Some additional advantages include local control of local data, accessibility to remote data, increased capacity, incremental growth, data availability, efficiency of storage, flexibility and cost effectiveness.

There are also some potential problems or disadvantages associated with distributed database such as the complexity of implementation - but this is the vendor's problem. Some additional potential problems include the problem of how to design systems for distributed environments, the complexity of administration and control, the impact on local operations, the political problems dealing with the ownership and protection of the data and the possibility of a node or line "crash". In addition, solutions that are appropriate in a centralized environment may frequently not be appropriate with distributed systems.

Chris Date is the author of 12 rules of a distributed database system. He begins with the fundamental principle or "rule zero" that states, "To the user, a distributed database system should look exactly like a nondistributed system". The subsidiary rules follow:

1. Local autonomy
2. No reliance on a central site
3. Continuous operation
4. Location independence
5. Fragmentation independence
6. Replication independence
7. Distributed query processing
8. Distributed transaction management
9. Hardware independence
10. Operating system independence
11. Network independence
12. DBMS independence

These rules are fairly self explanatory and will not be expanded in this paper. No current DBMS vendor adheres to all of these rules. A few vendors claim adherence to rules 1-8 and almost none to rules 9-12.

Hewlett-Packard is using a phased approach for developing a distributed database management system that will be rolled out in the early 1990's. The following is a summary of that approach:

* REMOTE DATABASE ACCESS (ALLBASE/NET) - Program can read/update a remote DB without coding for communication and remote processes.
* FOREIGN DATABASE ACCESS (e.g. IBM's DB2) - Program can access multiple vendors databases without coding for DBMS differences.
* DISTRIBUTED UPDATES WITH TRANSACTION MANAGEMENT - Enhanced transaction management to support updates to multiple DB environments.
* MULTI-REMOTE DATABASE ACCESS - Program can read and update more than one remote database at a time.
* SNAPSHOTS - Enables a user to copy all or part of a table from one database to another, optionally this table could be refreshed.
* DISTRIBUTED JOINS - Enables users to "join" data which resides in separate databases.
* PARALLEL QUERY EXECUTION - Ability to execute multiple queries to multiple databases at the same time.
* REPLICATED DATA - Enhanced availability and performance through multiple copies of data, with automatic synchronization of copies.
* PARTITIONING - Enhanced performance and availability through partitioned tables.

Some of the issues regarding distributed database that will have to be addressed in the 1990's include:

* Distributed query optimization and decomposition
* Fragmentation, recombination and optimizability of data

* 2-phase commit and recovery
* Referential integrity across sites
* Management of replicated/partitioned data
* Controlling authorized user access
* Update synchronization
* Degree of Transparency
* Flexibility to move data around the network
* Cost of mainframe vs. mini vs. micro MIPS
* Provision of foreign (non-HP) DBMS gateways (gateways are a way of processing data in a foreign DBMS or file system)

There are some factors that will help propel the distribution of data. These include company mergers, the downsizing of computers, the increased database needs and the general industry push toward distributed and cooperative processing. A company with distributed operations will gain competitive advantage through the support of distributed databases.

OBJECT-ORIENTED SYSTEMS

Each decade, one or two key advances emerge to change the practice of software development. Object-oriented systems and methods are rapidly entering the mainstream of software engineering and systems development. Leading consultants are heralding object-oriented approaches as one of the most important trends to affect businesses in the 1990's. But even among its strongest advocates, disputes abound over key issues, content, and definitions of the object-oriented approach. Object-oriented technologies are moving out of the academic world and into the business world.

With the object-oriented approach, processes revolve around the data, not the other way around. Using the traditional approach, programs are structured around data rather than procedures. For example, when using a traditional programming language, parameters are used to pass data structures and values between routines. The object-oriented approach attaches routines to data structures. In other words, the behavior of the data is kept with the data. This is called encapsulation.

An object-oriented programming language allows the programmer to define and manipulate objects. Some object-oriented programming languages are extensions of classical languages - C++ and ObjectiveC are in this category. Others are newer languages, e.g. Smalltalk and Eiffel. In addition, object-oriented extensions are being proposed for ANSI COBOL.

An object-oriented DBMS also supports the definition and manipulation of objects, plus providing the classic DBMS functions of persistent storage, transaction management, concurrency control, security, backup and recovery.

A "message" is an important concept used with the object-oriented approach. It is defined as a request sent to an object to change its state, or to return a result to the sender. Objects respond only to well-defined messages. The only information needed to use an object is knowledge of the messages it can receive. An object-oriented program is a flow of messages among cooperating objects.

Messages ensure the modularity of a system. To interact with any object, you only need to know what messages to respond to, not how the object is represented. Knowledge of how an operation is accomplished is of interest only to the programmer responsible for the definition of the object itself. Messages make an object's functionality available to other objects, while hiding the implementation details.

Maintaining and modifying software has been a real drain on programming resources. Maintenance programmers must understand a complex system well enough to fix its problems or enhance it. However, these programmers often did not develop the code, and often operate without adequate documentation or guidance. Changes often introduce new, unanticipated problems to the system. A programmer working with existing code must read and understand it; this may require a mental translation back into the original design specifications, which is extremely difficult for complex or poorly-coded systems. An additional concern is that programming languages allow unchecked access to data structure internals.

Objects can dramatically improve the problem of maintenance. Modularity and encapsulation limit interdependence, allowing changes that do not disrupt the rest of the system. Objects' natural organization make it easier to learn and understand relationships between parts of a system. The original programs are easier to write and debug and fewer errors occur. Programs read like designs, making changes clear and easy to make. Reuseability is an important advantage of using objects and libraries of these software components provide leverage. What has been written once need not be written again. Model features like inheritance allow existing components to be incrementally modified to suit changing needs. Together the representational advantage of encapsulation and the features of inheritance dramatically improve software development and can greatly improve programmer productivity.

Libraries of high-quality, tested software components will radically alter the way software is written. Software will routinely consist of a series of software components glued together. Application programming will no longer mean rushing to a text editor to begin coding. It will require understanding the capabilities and restrictions of available components, plus knowing how to combine them.

Some of the challenges facing object systems include the time it takes to learn about existing libraries of software components. Programmers also may resist accepting this new approach. Objects also consume more resources, however emerging 80486 PC's and RISC workstations will help to alleviate this problem. Applications with promising potential include: prototyping, user interfaces, graphics, telecommunications, geographic information systems, computer aided design (CAD), and computer aided manufacturing (CAM).

Object-oriented products are here today and the commercialization of object-oriented technology is increasing rapidly. Object database systems are currently viable for commercial projects and will be widely adopted by the mid 1990's.

To be successful, ODBMS technology must integrate well with existing data management systems. Using query interfaces similar to SQL will dramatically reduce retraining costs. Accessing data outside the ODBMS will allow users to use their existing applications in concert with new ones developed using an ODBMS.

Object-oriented environments herald the dawn of new programmings paradigms. Business people will be empowered to perform tasks that, in the past, required professional programmers. Programmers will be empowered to design complex applications in smaller, modular, more fool-proof pieces.

Neither end users nor application programmers will need to concern themselves with the mechanics of networking, peripheral support, or file handling. Object based architectures lend themselves to the creation of a much richer information environment. Digitized voice, music, images, video clips, and animation will begin to populate our information systems.

COOPERATIVE COMPUTING

The environment in which today's business must operate is changing quickly and becoming more complex. To meet the challenges produced by this changing environment, organizations need greater amounts of information to make the key decisions required for success. Keeping pace with the rapid change that is occurring means gathering information and making decisions faster than ever before.

As we look at the computer industry, we are about to embark upon the next revolution in computing. This is not a unique case. There have been multiple revolutions in the computing industry and computation in general, going back to the early days of mechanization, tabulation and so on, through the first computer mainframes. The mainframe was a very centralized processor, still oriented toward batch and was really a carry-over from punchcard tabulation systems. The next move was into mini-computers and distributed processing followed by the personal computer revolution.

This caused an explosion of workstations and personal computers on people's desks, which fueled distributed processing and distributed computing. Then came the communications revolution, where the objective was to integrate all this computing power in the corporation in a way that moves information around and enables different types of devices to participate in the solution of business problems. This has become the basis or launching point for the next revolution of cooperative computing.

Cooperative computing is the notion of tying together all the information computation resources in a corporation into a single entity, and making all those things interact in some efficient manner, transparent to the end-user in such a way that each user has access to all the information computation resources in the network as though they were local to the user's workstation.

NewWave Computing is Hewlett-Packard's implementation of the industry trend towards cooperative computing. It includes HP's vision of the future of computing: a network of heterogeneous computers that can work together to solve a single problem and are extremely easy to use.

To support cooperative computing requires a strategy that addresses the technological implementation needs. The NewWave Computing Architecture brings together systems and servers, easy to use workstations, industry standard networking and perhaps most importantly, the integration of the three through innovative software.

This paper has addressed many of the important technologies that are essential to the cooperative computing environment. ALLBASE/SQL is HP's strategic relational DBMS of the future for cooperative computing. ALLBASE/SQL runs under both MPE and HP-UX. Application development environments for both of these platforms include powerful tools from both HP and Third Party vendors. ALLBASE/SQL will also be the basis for distributed database technology in the future. ALLBASE/NET is the first phase of HP's distributed database technology providing remote data access and uses HP's networking capabilities.

HP ALLBASE/Turbo CONNECT provides coexistence between ALLBASE/SQL and TurboIMAGE by allowing ALLBASE/SQL applications to read TurboIMAGE databases. With ALLBASE/Turbo Connect, customers can preserve and leverage investments in TurboIMAGE/XL applications while reaping the benefits of relational technology through ALLBASE/SQL. A single industry-standard SQL interface allows access to both types of data.

ALLBASE/DB2 CONNECT allows ALLBASE/SQL to access (read and write) DB2 databases on IBM mainframes. Access to DB2 from both PCs and HP3000s will be supported from ISQL (Interactive SQL) and two PC-based products: Information Access and NewWave Access. ALLBASE/DB2 CONNECT is the flagship product by which we are launching connectivity to non-HP database servers, and extending the PC-HP3000 environment to include IBM mainframes.

Object-orientation is a very familiar concept at Hewlett-Packard. The concept of "objects" in the HP NewWave environment is similar to the object-oriented capabilities described in this paper. The Object Management Facility (OMF) is one of the main components of the HP NewWave environment. The OMF tracks all data in the PC, whether it be text, graphics, spreadsheets, scanned images, even voice. These objects are represented as icons which can be combined into compound documents containing different types of data.

The OMF allows users to create "hot-links". Hot-links allow users to share data between different reports, memos, even file folders. When data is changed in one place it is updated automatically in other shared documents throughout the system.

The OMF binds applications and data together to form "objects". By double-clicking on an object, you can simultaneously load the application and call up the desired file, eliminating the need to find file names and their directories.

In addition, Hewlett-Packard announced HP OpenODB in October, 1991. This object-oriented DBMS (ODBMS) product is based upon the Iris ODBMS prototype developed by HP Labs, starting in 1984.

HP OpenODB is an object-oriented model built on top of HP's relational DBMS storage manager (ALLBASE/SQL). As such, it supports all the DBMS features users have come to expect (e.g. security, transaction management, declarative query language, and data integrity). In addition, HP OpenODB's evolutionarily framework allows users to progress naturally to object-oriented technology while still accessing and using existing applications and their data.

HP OpenODB's ability to be ported to any vendor's platform makes it the most open, general-purposed ODBMS on the market today -- and easily two to three years ahead of the rest of the industry.

SUMMARY

The important technologies briefly addressed in this paper: SQL, distributed database management systems and object-oriented systems are extremely important to the future of the cooperative computing environment. HP's NewWave Computing strategy is focused upon helping our customers meet the challenges of today's changing business environment. We believe that these technologies and strategy, coupled with HP's commitment to industry standards and reputation for high quality, reliable systems, can provide our customers with the solutions they will need into the next generations.

What are we to do with SQL?

Codd himself does not think that SQL is a particularly outstanding ambassador of the relational model. On the contrary, he says on page 444 of his book, "Vendors, however, are forging ahead with both 'products on top' and 'distributed RDBMS products', disregarding errors in present relational DBMS products. All the evidence indicates that they will continue to do so. An inevitable result is that existing errors will become more difficult to fix, because more products and more users will be affected. Over time, the defects and deficiencies in the present versions of SQL will become totally embedded in relational DBMS products. It is important to be aware that, first, the language SQL is not part of the relational model. Second, the defects and deficiencies in SQL correspond closely to the various departures of SQL from the relational model."

During lunch, immediately after our public conversation, Dr. Codd and I discussed the SQL issue. As an example of the confusion, he mentioned that, during a visit he paid to one of the various SQL-standardizing committees, the members of the committee agreed on only one thing: they agreed to disagree with Codd.

Regardless of SQL's weaknesses, it is obvious that it is today's lingua franca for databases. The desire for a single, common language is nothing new. For instance, scientists have spoken all kinds of native languages, yet they have felt the strong need to inter-communicate. One solution would have been for each scientist to learn all of the languages spoken by everybody else. Because this would have been unlikely, scientists "agreed" (voluntarily or not) on *some* common language. In this manner, each scientist had to speak at most two languages, the scientist's native language *and* the common language of the day.

There are two important points about common languages:

1. The existence of a common language does *not* preclude the existence of "native" languages.

2. A common language is not forever, as it depends on political factors.

Because political factors are constantly shifting, several common languages have come and gone in the Western scientific community: Greek, Roman, German, French, English. Today, English is the common language for computers. As a highly-structured subset of English, SQL appears to be the emerging common language for database inter-communication.

Given these facts, we might as well learn SQL, even if we don't approve of it. And we might as well teach SQL to our favorite database management system.

Hewlett-Packard's ALLBASE Idea to the rescue

The beauty of Hewlett-Packard's ALLBASE concept is its *inclusive* quality. HP is making significant progress toward fulfilling this ALLBASE promise. For instance, ALLBASE/Turbo CONNECT currently allows SQL read-only access to IMAGE

AUDIT AND CONTROL IMPLICATIONS OF THE OSI SEVEN LAYER MODEL

Jeff Gibbs
Hewlett-Packard
3000 Hanover Street MS20BK
Palo Alto, CA 94304
415 857 6984

INTRODUCTION

In today's complex information technology environment, security administrators and auditors are often asked the questions: "How do I know when I have enough security controls? How do I know there aren't any holes in my security controls? And how can I detect when I have redundant security controls in place which may be reducing the productivity of my operation?" Typically, organizations may find themselves with multiple layers of passwords and still may be uncertain that they have provided adequate protection for their information technology resources. The establishment of adequate security for information technology resources is increasingly becoming a complex task due to the rapid proliferation of new system and network facilities. How can security administrators and auditors know when they have the right controls in place (neither too many nor too few)?

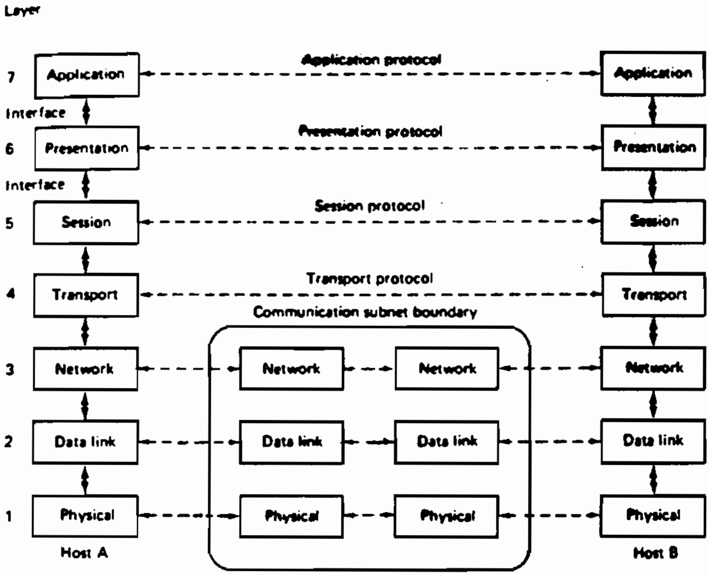
OSI SEVEN LAYER MODEL

One approach is to use the OSI Seven Layer Reference Model as the basis for IT security and audit strategy. The Open Systems Interconnection (OSI) model is emerging as a worldwide communication standard. The protocols of the OSI model will enable data to be exchanged between computer systems of different vendors. Increasingly, various organizations will be using standard formats and protocols for communicating messages and standardized communication paths for transmitting these messages. Most major computer manufacturers and communications companies have issued statements pledging conformance with these standards.

How does the model work? Basically, the seven layers of this model are as portrayed in Exhibit I with each layer of the communication in one environment sending protocol-based messages to its "opposite number" in another environment. One appealing characteristic of this model is the inherent "LAYERING" of sequential functions and controls. To communicate or interoperate in a commuications environment, many messages must be performed by the message sender, the message receiver and the network provider. Groups of related functions can be placed in categories or layers for planning purposes. The sequential functions that must be performed have been grouped into the conceptual framework which is called the OSI Seven Layer Reference Model.

EXHIBIT I

THE OSI REFERENCE MODEL



Each layer contains functional activity groups that specify and define a separate aspect of the telecommunications process. Each layer adds information to the message by means of protocols. That information provides the formatting instructions by which the message, when received, can be translated into a usable form.

The lower four layers concern the physical aspects of communications. For instance, physical connections, error control of transmission, and destination control are all achieved in these layers. These four lower layers provide for the reliable and accurate transfer of messages through the network to the trading partner.

The upper layers five through seven provide for the interpretation of the message by the message recipient. Only after interpretation is the message usable to the receiving organization. Without the higher layer functions, transmitted messages would be meaningless for business purposes.

For the purposes of this paper, these seven layers need to be supplemented by an additional layer which is what we think of as application systems running on host computers. For ease of discussion, this additional layer will be referred to as the Business Application layer or Layer "BA".

LAYERS AND MESSAGE HEADERS

Each layer contains sequential activities that must be performed consistently for messaging to take place successfully. The layers represent functional groups of linked activities. The layers must be correctly linked to result in a successful communication.

To portray the communication process, assume that a message is generated and is prepared for transmission, as follows:

MESSAGE

The message may have perhaps been generated by the entity's accounts payable system. Headers are now added to the message at each layer by the sending entity's communication software and hardware.

At each functional layer, information concerning protocols is attached in the form of headers for communication interoperability and control purposes. The first header (Hdr7) attached to the message occurs at Layer 7, the interface to the entity's business application. Then from Layer 6 down through Layer 2, headers (Hdr6 through Hdr2) are added for message control purposes. At Layer 1 the message and its headers are physically passed into the communications network for transfer to the entity's trading partner. The headers are messages that contain information such as name and address of receiving entity, types of security controls utilized, hash totals, type of encryption algorithm utilized, and data formats.

The communication that is submitted to the network includes the transaction message and its sequenced control headers, as follows:

HDr2	HDr3	HDr4	HDr5	HDr6	HDr7	MESSAGE

The reverse process takes place with the message receiver. The headers are sequentially read and the control information is used to interpret the message and to check security and integrity.

Each of these layers has certain specific security exposures and available controls. The security exposures at Layers 1-3 are of the nature of unauthorized physical connections (eg. wiretaps) while exposures at higher layers are more of the nature of inappropriate logical access (eg. logons).

Similarly, the controls at each layer are somewhat different. At Layers 7 and above, the appropriate controls tend to be such things as password and user authentication while at the lower layers, appropriate controls are such things as the shielding of wires and connections.

Error messages can be generated and logged at each level to provide an audit trail. As an example, Layer 2 framing activity has control features that can result in the generation of error codes when there are transmission problems.

INDEPENDENCE, TRANSPARENCY, AND DOMINANCE

Three important features of this layering approach are independence, transparency and dominance. If there is a change in a particular layer's protocols, and all the involved network parties cooperate by implementing the new rules (protocols), the other functions or layers can be unaffected. This demonstrates the concept of the "independence" of one layer from the other layers. This concept allows, for example, for the changing of a piece of equipment used by a particular layer without having to change operations at other layers. Also, a security control can be added at a particular layer without requiring changes at other layers.

This layering principle allows functions performed at one layer to be transparent to other layers. For instance, assume a protocol change is implemented at Layer 7. This control's operation does not depend on whether the message is transmitted via wire cables, optical fiber, or microwave (Layer 1). Thus, the transmission means and paths are "transparent" to Layer 7. The same concept of transparency occurs with the mailing of a letter. The Sender and Receiver are not concerned with the physical path of the letter. They are concerned that the sealed letter is delivered in a timely manner.

This transparency principle has an important corollary, as follows:

SECURITY CONTROLS APPLIED AT A PARTICULAR LAYER WILL BE EFFECTIVE IN CONTROLLING EXPOSURES DERIVING FROM LOWER LAYERS.

Following this principla, controls applied at a higher layer tend to dominate or take precedence over, exposures coming from a lower layer. For example, if text is encrypted at Layer 6, it will then be relatively immune to security exposures at the five lower layers. Similarly, password controls applied at the Business Application layer should effectively preclude the possibility of inappropirate access regardless of what physical transport medium is utilized.

CONTROL IMPLICATIONS

Why then couldn't entities consistently rely on Layer 7 and above controls to preserve the security of their information technology resources? After all, in the final analysis, isn't it really the application data itself which we seek to protect from inappropriate access? The following are some important considerations:

1. By merely controlling security at Layers 7 and BA, we are, in effect, allowing free access to our transport and network resources. These resources are not without cost to entities and it is doubtful that many entities would like to serve as public access utilities carrying other entity's messaging traffic at no charge!

2. A strategy which relies on controls at Layer 7 and BA is, by nature, decentralized in its application. The lower layers are, in effect, the glue which holds our information technology resources together. By applying controls at these lower layers, we can encourage a more consistent application and avoid a "weak link" phenomena. That is, in the decentralized environments which characterize Layers 7 and above, if individual applications are unsecured, it may create an exposure which could effect other networked resources as well.

3. On the other hand, many data security analysts question the feasibility of providing foolproof controls at Layers 1-4 due to the multiplicity of potential access points for networked resources.

AUDIT IMPLICATIONS

How can security administrators and auditors utilize these concepts to evaluate the security of IT resources? One possibility is suggested in the matrix illustrated in Exhibit II. In this matrix, some typical security objectives are listed on the vertical axis. The horizontal axis should contain all the controls relied upon by the entity to meet these control objectives. This controls axis is organized by OSI Layer with Layer BA being added for completeness.

EXHIBIT II

OSI SEVEN LAYER CONTROL MATRIX

CONTROLS

SECURITY OBJECTIVES	BUSINESS APPLICATION LAYER			-- LAYER 7 --		-- LAYER 6 --	-- LAYER 5 --	---- Layers 1-4 ----			EVAL
	Pass-words	User Caps.	Edit Checks	Logging	Encryp-tion	Data Compress.	Record Format	Cable Shield.	Address Tables	Sequence Numbers	
EFFECTIVE USER ACCESS SCREENING											
DETECTION AND FOLLOW-UP OF UNAUTHORIZED ACCESS ATTEMPTS											
DATA CONFIDENTIALITY MAINTAINED											
ONLY AUTHORIZED CHANGES MADE TO PROGRAM CODE											
ONLY AUTHORIZED CHANGES MADE TO APPLICATION DATA											
EFFECTIVE PHYSICAL ACCESS CONTROL											
EFFECTIVE VIRUS DETECTION AND CONTROL											

KEY: CONTROL IS ADEQUATE PART. ADEQUATE INADEQUATE N/A

In using this matrix, the security administrator and the auditor should fill in each cell with the degree to which the entity relies on a particular control to meet its security objectives. Keeping in mind the relationships mentioned above, if the entity intends to rely on Layer 7 or Layer BA controls, it may obviate the need for controls at a lower layer. However, as also indicated above, to be effective, these higher layer controls would need to be consistently applied across all application systems in the entity's environment.

Once the entity's security strategy has been determined, the matrix can then be used as the basis for detailed audit planning. Critical dependencies could be identified utilizing this approach, thus guiding the auditor to the specific controls which would need to be tested.

This approach provides several advantages in planning and auditing IT security:

1. It assists security administrators and auditors in clarifying the entity's security strategy.

2. It assists the auditor in focusing field testing on critical controls.

3. It can serve as a vehicle for communicating control issues and reaching agreement on necessary corrective action.

Most importantly, the use of this type of matrix allows security administrators and auditors to answer the important question posed at the beginning of this article: "How much is enough?" An examination of the controls planned for each security objective along with a testing of these controls should demonstrate conclusively the adequacy of these controls for the intended purpose. In addition, by reviewing the verticle columns for each control, the cost-benefit of particular controls may be more easily determined.

Bridging Islands of Information:

An Overview of Hewlett-Packard's

Distributed Application Architecture

Michael J. Mathews

Abstract

As we move into the Information Age of the 1990's, and computing technology evolves to become Information Technology, we will witness the development of a well integrated computing environment that is highly distributed, based on portable, inter-operating components from many different vendors. To make this vision a reality, and in so doing, deliver on the vast potential of computing technology as a useful business tool, a common and consistent application environment must be supported by all of the component vendors. A broad architectural framework, widely accepted and based on open systems and standards is the answer.

Hewlett-Packard's Distributed Application Architecture describes a cooperative, distributed, object-based system designed to: *1)allow users of a computer network to access information, applications and services as well as to exchange information with others through a single, consistent user environment; and 2) allow software developers to create new information, applications and services rapidly by taking advantage of those that already exist.*

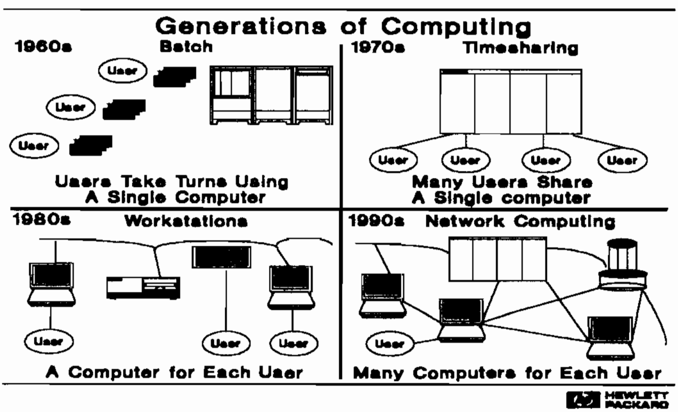
1. Introduction

Over the past 30 years computing technology has evolved through several generations of user/system relationships. In the 1960's the batch oriented systems required users to take turns using a single system. Each application ran uninterrupted in its own computing environment, to be followed by the next application ready to run. Throughput was limited by the speed and capacity of the systems, so demand for computing resources was primarily limited to large data-processing intensive repetitive tasks such as accounting and order processing. The output was limited information in a fixed form based on historical business practices. It was useful in monitoring business versus being useful in making business decisions.

In the 1970's timesharing systems allowed many users to share a single system. Individuals now had direct access to some data and to the resources to process it. The output of much of this processing was useful information that supported the administrative business decisions of those individuals. In other cases the processing supported engineering, manufacturing and other functions involved in the creation of goods and services.

In the 1980's individual workstations and networks made it possible for each user to have their own computer. The expanding need for more computing power to access and use information drove the market to accept the use of individual computers for each user. Though throughput increased dramatically, with more information becoming available to support increasingly complex business environments, the information could not be easily integrated or shared. In effect, islands of information had been created offering only limited inter-operability and inter-action.

As the 1990's unfold we will experience the power and flexibility of network computing. Computer users will continue to work with an individual workstation, but they will now be well integrated with other workers in their department, location and company. They will have direct access to sources of information needed to support their function. Many (different) computers will be used to support the various aspects of their tasks, but without their knowledge or direct intervention. In effect, the islands of information will be inter-connected to form an information infrastructure, enabling the access, exchange and sharing of information, resources and services across the entire network.



2. Challenges of Information Technology Delivery

To deliver value in the Information Age, Information Technology will have to meet the following challenges head-on, and overcome them.

- <u>Easy Information Sharing with Dispersed Teams</u>

  Individuals and teams will cooperate across time and space to achieve a common goal. They will have to have access to the information and resources required to be successful. They will also need to work together on common parts of the task without being collocated. Dynamic exchange of ideas will be facilitated electronically, with dynamic feedback to augment and modify the process. Virtual presence will replace physical presence as the workstation becomes a conduit to the rest of the world.

- <u>Arbitrary Integration of Multiple Media Types</u>

  To be useful, information will have to be delivered in the media form in which it offers the greatest value. This may be textual, tabular, or graphical, all common today. It may also be animated, audio- or video-based, and as other forms of media are developed they will have to be used where appropriate. the key to success is the ability to convey information in a natural, easily assimilated form without loss of content or meaning. The crucial goal will be to deliver the correct information as efficiently as possible.

- <u>Timely Delivery of End-User Solutions</u>

  End-user solutions will be tailored to meet the individual's requirements. The components of a task will be combined and customized to enable the individual to perform their task with greater productivity and accuracy. Components of the task will be added, changed or deleted without impacting the other components of the task. Developers will create new components re-using and/or modifying parts or all of existing components. End-users will become "developers" by dynamically assembling task-oriented solutions out of the variety of components they have available to them.

- <u>Automation of Routine User Tasks</u>

  Repetitive tasks will be automated to free the end-user from having to 'drive' the system. Captured by recording actions or following directions in a script, simple tasks will be able to play back the actions necessary to complete the task. Information retrieval, production of reports, personal administration, to name just a few, will be handled by the end-user's surrogate or 'agent'. More sophisticated task automation will be possible using artificial intelligence driven by knowledge base supported inference engines.

- <u>Approachable Systems for Today's Apprehensive Non-users</u>

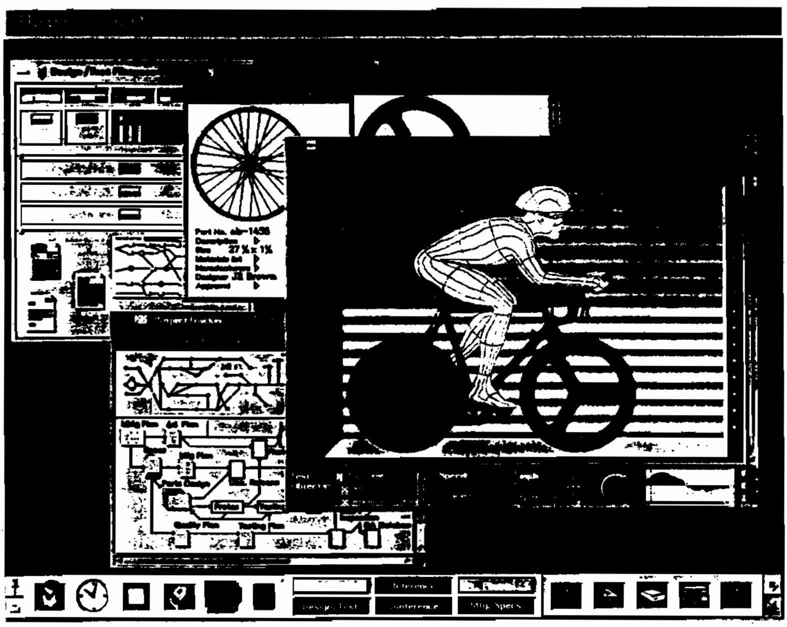
  As computer technology becomes an indispensable tool in our daily lives, and thereby more pervasive in our society, it must also become more usable and approachable. It should be intuitive - setting and meeting expectations of use. It should be consistent - re-enforcing the model of use. It should be more flexible - allowing new components to "plug and play" with existing components. Today's non-users will have to be provided with an environment that meets their requirements before their use of an information infrastructure can become indispensable to them.

3. Visualizing the Future

Sometimes a picture can help to illustrate the example far more effectively, so I have put together four scenarios that might be common place in the future.

- <u>Helping You Do It Your Way (in the engineering world)</u>

  In this example I have selected the typical engineering environment to show the use of object-orientation applied to the problems and tasks a design engineer deals with. The environment is customized to this engineer's specific functional requirements. The task is to design a wheel for a bicycle. The components of the wheel are all described as objects with their own functionality and attributes. Each wheel design inherits aspects of its constituents, and in turn exhibits functionality and attributes. Testing can be performed on each wheel design (in a computer simulated wind tunnel) with components being dynamically changed as the engineer decides. Behind the wind tunnel window a project tracking flow chart shows the effect of design changes on the project plan. This is also updated dynamically by changes that occur in related design spaces.

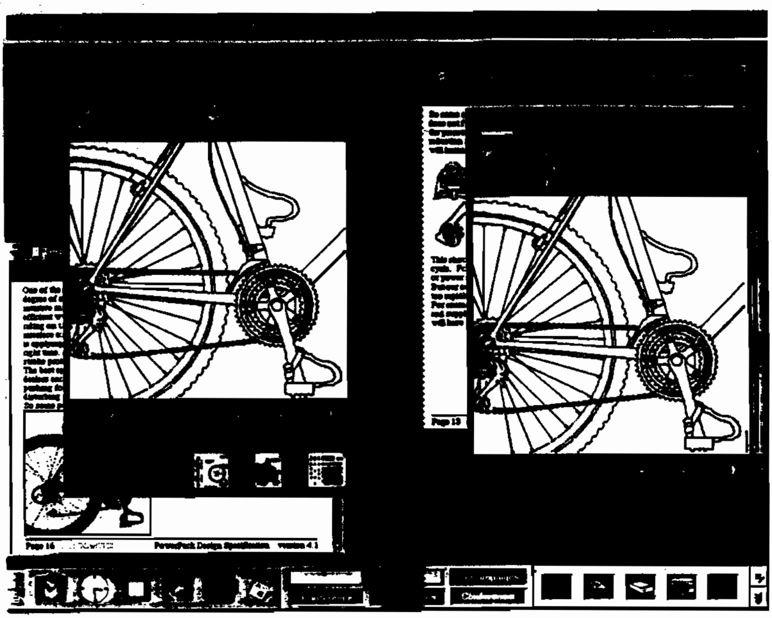


- <u>Helping You Do It Your Way (in the office)</u>

  This environment depends a great deal on access to the right information. The manager makes extensive use of "agent" technology as well as object-orientation. The agent scans news sources and assembles custom newspaper abstracts prioritized according to the manager's interests. A full article can be obtained, and either abstracts or articles can be used to tell the agent to find similar material from other dates or sources. The agent is able to find, filter, organize and present the information, thus freeing up the manager for more productive work. The agent also files the information and can retrieve it on request. A key aspect of this technology is its ability to "learn" based on the end-user's feedback and requests. Another key aspect is the end-user's "natural" interface to the agent - in this example by hand-written natural language (an alternative would be voice).

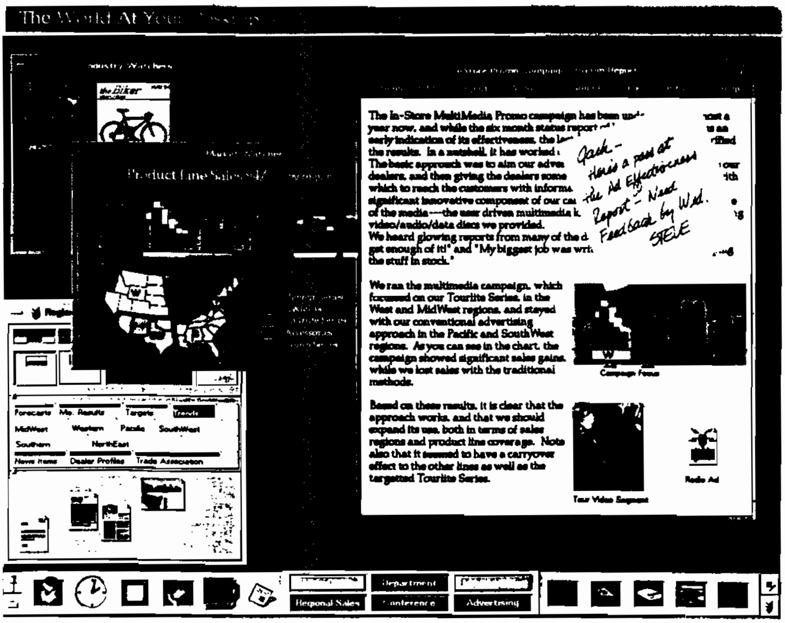
- <u>Working Effectively With Others</u>

  Here we are showing how the architecture enables "virtual presence" of team members assembled to resolve a specific problem. An electronic meeting has been called by a factory support engineer dealing with a customer hotsite. The participants in the meeting include the field support engineer, the original design engineer for the product at issue, and the manufacturing manager for that same product. Their objective is to bring together their relevant expertise supported by the information they have to understand the nature of the problem and decide if it justifies delaying a major production run about to start. Each participant joins the meeting through his/her workstation. They share their information into the meeting forum, and each can dynamically annotate it during the meeting. This information is presented in several media forms including textual tabular, graphical, voice and video. The essence of this illustration is the combination of the right people with the right information to solve a problem in the most efficient and timely manner.



- <u>The World At Your Desktop</u>

  My last example shows the work environment of a product or brand manager in a marketing organization. To properly manage and position a product in a highly diverse and widely distributed market the manager has to make use of rapidly changing market information gathered from external as well as internal sources. The task in this case is to analyze a series of promotions that have been running in different regions to assess market penetration. Some key aspects of this environment are the need to establish persistent links between internal and external objects, object-containment where objects (possibly remote) are contained in other objects, multimedia support where the TV video ad and the radio voice ad are part of the report. The report will be sent to region as well as corporate marketing executives, so the manager has distributed it within his department for review with a "post-it" note object asking for feedback.



4. The Distributed Application Architecture

The Distributed Application Architecture (DAA) describes a cooperative, distributed, object-based system. Systems built upon the DAA are intended to operate across a heterogeneous collection of machines.

The purpose of the DAA is *to allow users of a computer network to access information, applications and services as well as to exchange information with others through a single, consistent user environment. The DAA will allow developers of all kinds to create new information, applications and services rapidly by taking advantage of those that already exist.*

The architecture is designed to provide specifications for the implementation of systems that will build upon the success of, and expand on, the initial implementation of NewWave on the PC/DOS platform. In the DAA environment it is possible to implement systems that take full advantage of the greater capabilities of operating systems such as UNIX.

A key focus of the DAA is cooperation among its objects to provide solutions to the end-user. A design based on objects encourages both users and programmers to focus on <u>information</u> to be used and <u>operations</u> to be performed to achieve the desired solution, rather than on machines, application programs, storage devices and other elements of the underlying technology. Basing the system on objects also enhances the system's extensibility and establishes the framework for a consistent (user's) conceptual model of use.

4.1 Features and Benefits

One of the primary goals of the DAA is to provide a basis upon which users and applications can cooperate with one another. The architecture is designed to support the realization of environments in which users and applications are integrated, in all combinations: users with applications, applications with other applications, and users with other users.

To achieve integration there are three important aspects of applications to be modeled in a DAA environment. These are applications as information managers, applications as service providers, and applications as clients or users of information managers and service providers. Clients are defined to include both end users and pieces of software which utilize the capabilities embodied in information and service providers.

Some of the benefits of integration that we are seeking include: allowing diverse pieces of information to be combined and used together; allowing tools to be used in combination with one another in useful ways, often unanticipated by the developers; and allowing tools to be used flexibly to process diverse pieces of information. In the model, an object oriented paradigm is used as the basis for integration.

Users are clients of the integrated services in the environment, but clients can also be other objects. Such objects can act as autonomous agents which automate, monitor and control the flow of processing in the environment. In effect, objects (programs) can themselves simulate or mimic the actions of end users (humans).

In summary the DAA provides the following features:

Non-Proprietary Nature. Implementation of the DAA does not require Hewlett-Packard proprietary operating system or hardware technologies. It will be implemented on industry standard operating systems, initially UNIX.

Open System. The DAA is an open architecture that can be licensed by anyone. Independent software vendors will be encouraged to write applications and common user facilities for the architecture.

Support for Distribution. The DAA facilitates the development of applications that run on multiple hardware platforms connected by both local area networks (LANs) and wide area networks (WANs).

Support for Interoperation and Integration Across Heterogeneous Platforms. Applications written to the DAA will interoperate across different computer systems implementing the architecture. A distinctive feature of the DAA is that the connections typically can be made by end users or system integrators, not just by developers.

Support for Software Agents. As well as facilitating the development of object-oriented applications, the DAA will also include support for the concept of the software Agent that can be used to monitor and control objects.

As a result systems built using the DAA are:

Easy to Use => Reduced User Support Costs. Ease of use is the primary key to making people productive. DAA systems are intuitive and therefore easy to learn and recall. The solutions should be more accessible for more people than is true with current systems. People will not have to learn arcane computer jargon and special procedures to do their work. The system will support, not impede, the user in doing their work. User interface consistency will allow people to leverage training in their operation.

Adaptable => Reduced Software Development and Maintenance Costs. DAA systems are flexible and easily adaptable to changes in the problems being solved. The architecture encourages the development of modular systems. This is so that capabilities which are common to many solutions can be shared, which will also avoid unnecessary costs and incompatibilities.

Cooperative => Increased Team Productivity. Software written to the DAA enables cooperation between users and also eases collaboration with independent pieces of software through its adoption and promotion of standards for interoperation.

Automated => Increased Personal Productivity. Agents in DAA systems provide for useful and effective automation of repetitive tasks. Agents also make compliance with procedures set up by an organization, both automatic and minimally oppressive to the user. Agents are powerful tools for seamlessly integrating and embedding the functionality of multiple, diverse objects within and across heterogeneous environments.

5. Underlying Principles

The DAA rests on a set of principles which influence the way its objects behave and must be managed. Following are some of the most important.

Distribution: A DAA system is distributed across a number of machines which are connected by a network.

Heterogeneity: The machines that comprise a DAA system may be from different manufacturers, may use different processors with different binary code formats and ways of representing data, and may have different operating systems, window managers, networks and so on. One of the main duties of the DAA infrastructure is to shield application developers from the data communications network and differences in platforms.

Independently-Developed and Installed Software: The software which supports DAA objects will come from many vendors. Different machines in the system will have different sets of applications (and different versions) available. Not all applications or versions are required to be on all machines.

Run-time Integration: The DAA describes a system to support interworking among independently-developed objects. Connections will normally be made by end users or system integrators, as their needs demand.

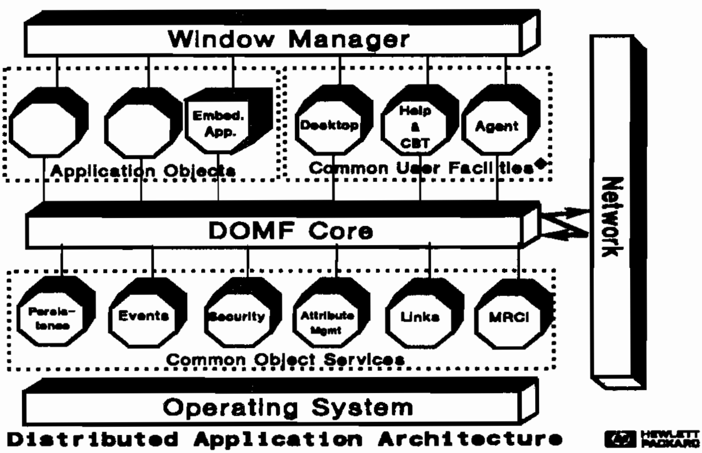
Multiple Users: The architecture allows objects to be used simultaneously by multiple (human) users, as well as by multiple other objects operating on behalf of various users.

Integration with existing applications: Conversion to DAA will take time. The architecture facilitates conversion, and supports coexistence with existing applications during the conversion period.

Relationship to other Object-Oriented Technologies: The architecture supports the integration of other object-oriented technologies (e.g. object-oriented data bases and programming languages).

6. Architecture Model

The diagram presented in the figure below is a top level picture of the major software components in the Distributed Application Architecture. It illustrates these components in relation to each other, and serves as the basis for describing components of the architecture.



A complete system based on the DAA will include both components which supply *services* provided as part of the infrastructure, and a set of conventions or *policies* defining how other components are to interact with the provided services and with each other.

The application architecture is centered around the concept of the DAA *object*, an abstract data object which interworks properly with the DAA infrastructure and subscribes to the appropriate DAA policies.

The Distributed OMF (Distributed Object Management Facility) is the component of the DAA infrastructure that manages the existence and execution of objects, along with the relationships and interactions between objects. Because the DAA is intended to support a wide range of application areas, certain features which are needed in some contexts will not fit in others. Consequently the architecture is modular. The Distributed OMF is the *core* layer of the architecture, providing basic functionality common to any DAA-based system. The concepts embodied in OMG's Object Request Broker standard are derived from the DOMF specifications submitted to the OMG.

Common Object Services are the object-management *policies* and *supporting services*. Some *policies* (and their *supporting services*) will have broad applicability, while others will be more narrowly defined. This modular approach to the architecture allows considerable flexibility in defining different kinds of systems. The architecture is not dependent on any specific operating system or network transport.

Common User Facilities is the collection of objects widely available on (most) DAA systems. The Desktop is the user interface to the Distributed OMF. It is itself an object which is a customizable component of the architecture. It can be used to manipulate objects both within and outside the object-based environment. It communicates with the end user via the window manager. The architecture is not dependent on any specific window manager technology.

Agent is the general term for services (objects) that can be constructed to monitor and control other objects, and the interactions between them. Agents are used to automate repetitive tasks, to automate (and facilitate compliance with) standard processes, and to create new solutions by integrating and combining existing objects.

Help & CBT (Computer Based Training) are specific examples of Common User Facility objects. Other examples of such objects include a print (spool) object and a mail object.

Application Objects are the objects that are used to accomplish specific end user functional tasks. Usually consisting of a number of discreet DAA objects, they may have a specific focus (such as an accounting application or a CAD application), or be more generalized (such as word processing or spreadsheet applications). Applications can be implemented to take advantage of any level of distributed support provided by the DAA. Many alternative implementation models exist including: *embedding, monolithic, presentation/semantic, client/server,* and *distributed display manager.*

7. Basic Concepts

Within the DAA environment the following concepts are used when referring to objects and their characteristics.

7.1 Objects

An *object* comprises some state and a set of defined behaviors. The state is represented by data maintained by the object. The behavior is specified in terms of operations the object can perform, and is realized by executable code. Conceptually the data and code are inextricably bound together. Objects communicate with each other by sending requests via messages. An object can provide services to clients (typically other objects) by means of the operations it can perform. Other forms of messages include event notification.

An object's external specification provides an *abstract* description of the object's behavior (services it can perform) and an abstract state model. The state model provides a way of describing the effects of previous operations on future behavior. The *abstract* description is central to the (human) user's concept of what the object is, and usually presents the object in familiar terms. The supported operations provide the only way another object can access or alter the object's state. This hiding of the actual details of implementation is called encapsulation. An object's state may be simple or complex - a single value or a complete design drawing (itself consisting of many objects). Similarly the functionality of an object's operations can be lean or rich - such as a simple add operation or a complex print operation. An *interface* characterizes an object's externally visible behavior, independent of platform. Part of an object's behavior specification is the description of the *interfaces* it presents. In general an interface is simply a set of operations the object can perform.

Objects which exhibit the same behavior in the same environment are described as being *instances* of the same *implementation class*. An object's implementation class completely characterizes its executable code and persistent state storage format.

Implementation *inheritance* within the DAA is supported in the form of callable libraries of standard operations (*library methods*) which free the object designer (application developer) from having to deal with the implementation of standard operations. *Composition* is a way of combining existing object designs to perform more complex tasks. This can be achieved in two ways: *instance composition* - the combination of existing objects; and *class composition*- the combination of library methods (applications).

7.2 Location

The DAA provides location-transparent access to objects. An originating object does not need to know the location of the target object in order to send it a message. The Distributed OMF ensures the correct delivery of all messages. Because all objects have unique identities, it is also possible for originally unrelated DAA systems to be joined together without risk.

Although the DAA supports location-transparent message delivery, it also recognizes that objects' locations may have practical effects on performance, communication costs, accessibility and reliability. To allow users to manage these effects, the DAA supports the concept of the *location* of an object. This allows systems administrators to control: object registration, object availability, and persistent storage, both for information about objects (*properties*) in the storage domain and for the objects' persistent state.

The DAA supports the moving of information in two ways. *Message interchange* is an interaction between two stationary objects, and can convey arbitrary control information or data. *Object interchange* is the process of moving or copying the state of one or more entire objects from one place to another. When an object is *moved* the object retains its original identity. When an object is *copied*, a new object, equivalent to the original, is created with a new identity. The architecture also supports the concept of *conversion*. This is the process of transferring [portions of] an original object to a new object which is different from the original. These capabilities enable the re-use of objects in various situations.

7.3 Relationships

To facilitate the integration of new objects the DAA uses *interface negotiation*. This is a protocol which an object can use to inquire about the operations supported by another object. The protocol also allows for the introduction of new interface definitions which were not included in the original system implementation.

To enable objects to retain persistent references to other objects the concept of *links* has been introduced. Links can be either uni-directional or bi-directional. Links have many uses: representing containment and passing information are the two most common. The DAA Link Manager reliably supports persistent links between objects on different systems.

7.4 Distributed Presentation

In a distributed environment users may interact with objects which are located on computers far away. To support this efficiently, the concept of *presentation* and *semantic* objects has been introduced. The role of the presentation object is to interact with the user to display information and receive input which is translated into commands to be executed by the semantic object. Thus the semantic object is the object the user appears to be using, even though it is physically located on some remote computer.

7.5 Conversion and Coexistence

To facilitate conversion from and coexistence with existing (non-object-oriented) applications the DAA uses the concept of *embedding*. This is a process for creating an object out of a non-object entity by wrapping it in an appropriate shell that enables communication between the original program code and the DAA. In essence the shell encapsulates the entity such that its capabilities are made available as operations to other objects.

8. Conclusion

This paper has discussed Hewlett-Packard's vision for computing in the 1990's, the software environment necessary to realize it, and the architectural description of this environment. The Distributed Application Architecture is an object-oriented software specification designed to support the creation and use of applications capable of inter-operating in a heterogeneous networked environment. Based on open-systems, industry-standard technologies, with no proprietary dependencies, it can be implemented on any underlying platform technologies that provide the required functionality. Systems based on this architecture will make it possible to build bridges between today's islands of information.

Paper #3801
Looking at UNIX Through MPE Eyes

Betsy Leight, Vice President
Operations Control Systems
560 San Antonio Road, Suite #106
Palo Alto, California 94306

Telephone: 415-493-4122

ABSTRACT

Converting from an MPE to an MPE and UNIX shop can be a trying experience at best. The operating system and the third-party software available are radically different, making the UNIX environment seem like foreign territory to the MPE veteran.

This paper helps ease the transition by providing comparison charts and programming checklists, as well as reviewing the migration saga of a twelve person team of programmers as they moved from MPE to the UNIX multi-user systems.

***HANDOUTS AVAILABLE AT TIME OF SESSION**

3802

DISASTER RECOVERY PLANNING (ON A BUDGET)

P. TROY JACKSON, CDP
SERV-AIR, INC.
LEXINGTON-BLUEGRASS ARMY DEPOT BLDG. #3
LEXINGTON, KENTUCKY 40511-5126
PHONE (606) 293-3744

Large companies have long been aware that the continued operation of their vital computer and networking systems must be insured. In order to provide this assurance, they have constructed elaborate and expensive disaster recovery schemes. Small and medium businesses are no less susceptible to a major catastrophe, but they have limited resources available to commit to such a plan. Frequently the managers of these smaller systems feel they must take an all or nothing approach, and due to extensive constraints, chose to do nothing. Alternatives do exist, and a cost effective plan can be developed to address at least the most probable situations.

As information systems expand to play a greater role in business, the impact of suddenly being deprived of these systems is of growing concern. This paper will address the issue of cost effective disaster recovery planning in a small to moderate business. Several areas will be discussed, including defining a "disaster" and "disaster recovery," the value of preventative measures, extraneous considerations, costs involved in developing and implementing a plan, and an actual scenario to demonstrate the feasibility of such a project.

> Today most, if not all, large Fortune 1000 financial institutions, banks, and insurance companies have disaster recovery plans (DRPs) to aid in the recovery process following data center disasters. The plans also help to reduce the liabilities and business interruption losses incurred. These large firms generally have the manpower, time, and money to develop their DRPs, and in addition may have "hot sites," where they can process their data should the sites sustain extensive damage.

> Smaller firms, which often use HP 3000s for distributed computing, may not have these "luxuries," and are thus much more vulnerable to data loss. In fact, the business and financial losses that can occur as a result of data center destruction can very quickly bring about the demise of an unprepared firm. (Robinson, 1988)

Developing a functional disaster recovery plan is always a demanding task, but to develop one that adheres to the budgetary constraints of a small to medium size business is almost an insurmountable challenge. "It is very hard to plan for something that, in all likelihood will not happen. Unless you have suffered a loss of data or an unplanned extended period of down time, you find yourself creating a plan that cannot even be thoroughly tested for validity." (Michel, 1989) It is, however, the small to medium facility that suffers the greatest potential loss in the event of a disaster.

> "Freeze! When disaster strikes, that's what many MIS directors have to do. They didn't buy an Uninterruptible Power Supply (UPS), didn't subscribe to disaster-recovery services, didn't even have a vague plan to cope with the many crises that can befall the corporate information resource. Often, budgetary constraints imposed by top management limit IS managers' options in a crisis." (Leibs, 1989)

DEFINITIONS

What Constitutes A "Disaster"?

The definition of a disaster varies tremendously. "Anything that interrupts normal processing to the extent that the client is severely impacted constitutes a disaster." (Magee, 1987) "It doesn't have to be a smoke-and-rubble situation. It may be as simple as a data communications problem or a down telephone line - anything that means your data center is simply unavailable." (Harris, 1989) "We define a disaster as a disruption of processing for an extended, undetermined period of time." (Business Recovery Systems, Inc., 1986)

What constitutes a disaster depends primarily on the peculiarities of the organization to which the discussion pertains. To a major corporation that relies on complex computer systems to function, even a few minutes of interruption could be considered a disaster. To a small business that utilizes a computer only to perform its in-house accounting, an interruption of several weeks could be merely an inconvenience. In many cases, the key factor in determining the magnitude of the problem is the time of day it occurred. Three hours of down time can range from devastating to unnoticed depending on the demand for the system at the time of the failure.

The determination of what constitutes a disaster is a key element of any recovery plan. If it is understated, essential considerations will be overlooked. If overstated, valuable resources could be misdirected. The person or persons responsible for the plan must carefully analyze the requirement of their specific situation.

> Management must realize that EDP professionals agree there are **no secure computers**. Management must realistically look at: legal obligations, cash flow maintenance, customer services, competitive advantages, production and distribution decisions, logistics and operations control, purchasing functions and vendor relationships, ongoing project control, branch or agency communications, personnel and union relations, and shareholder and public relations. Management must assess the importance of their information services to these facets of their business, then decide the type of effort that should be put into disaster backup of the EDP function. (Buckland, 1982)

It is an unfortunate reality that upper management will be more interested in the actual cost of a recovery plan than the potential cost of a disaster. It is the responsibility of the MIS manager to, not only develop the plan, but "sell" that plan to those who control the budget. This is a difficult task, but not impossible. One of the best ways to define a disaster for an individual situation is to outline the possible scenarios which might be encountered and develop mitigation activities.

What Is Disaster Recovery Planning?

Once again there are considerable differences of opinion as to how thorough and complete a plan should be. "The object of disaster planning is to develop a plan of action which, regardless of any event, will avoid catastrophe, limit damage and allow us to continue to operate under a contingency plan." (Leverentz, 1988) "Disaster recovery is the ability to successfully recover and/or restore all essential business functions within three days or less after a computer disaster." (Magee, 1987)

Existing plans range from complete, fully functional "hot sites" to periodic tape back-ups.

> Some options to recover operations require no outside service to provide backup. These options include returning to manual operations, owning a "redundant" or backup system at another location, or having a reciprocal agreement with another organization. Companies that require disaster recovery services pay an outside source for this service. Outside source options include hiring a service bureau to handle data processing needs until the system can be replaced or restored, using the vendor's facilities, contracting with a service that provides either off site or relocatable recovery facilities, or having a service provide an empty shell. (King, 1989)

Most professionals agree that the first step to successful recovery is to maintain some form of back-up scheme. "Our advice is: back up, back up, back up, and when in doubt, back up again." (Harris, 1989) "Backing up data is like buying life insurance: It's an unglamorous safety net you hope you will never have to use." (Scheier, 1989) Beyond agreeing that back-ups are essential, there is little agreement as to the extent or even the effectiveness of recovery plans.

How extensive a recovery plan is should be is directly proportional to the needs of the particular business developing the plan and the particular instance in which the plan is enacted. "The first step in effective disaster recovery planning is to pinpoint the problem that the planning is trying to solve." (Schlandweiler, 1989) For the crew of the space shuttle Atlantis, disaster recovery planning meant being prepared and trained to completely replace a defective CPU while in orbit around the earth. "They were not the first people to encounter computer troubles, nor were they the first humans in space. But the most recent space shuttle crew members were the first to conduct an outer-space CPU swap-out." (Ryan, 1989)

LIKELY DISASTERS

A great deal of difficulty can be avoided by only addressing the most likely scenarios as opposed to trying to provide for every contingency. While it is true that "any disaster is going to be different from what you anticipated," (Nielsen, 1986) that does not preclude the desirability of only planning for the most likely events.

Power Interruptions

By far the most common cause for computer systems failure is directly related to the electrical power. "Power disturbances have a mean time between failure (MTBF) ranging from 29 to 730 hours. In contrast, hardware MTBF ranges from 3,000 to 10,000 hours. Because the reliability of the electric utility power is 100 times less than that of the hardware, power has become the major cause of all computer maintenance problems." (Severinsky, 1989) Not only do power interruptions present problems, but power surges can severely disrupt service. "With the exception of lightning, one can't see a surge or its source with the naked eye.... And yet, circuits that repeatedly experience surges suffer damage that reduces equipment life and ultimately results in failure." (Hawley, 1989)

Other Disasters

In addition to power disturbances, the most likely disasters will vary somewhat due to location, but according to a recent survey the following are the most common: "fire (57%), floods (35%), and security violations (26%). (Harrison, 1988) Each individual situation must be examined to determine the best way to prepare for most probable events.

ADDITIONAL CONSTRAINTS

Frequently outside agencies dictate the magnitude or even the specific type of protection an organization must have. These constraints include government regulations, changing standards, and costs. In instituting a recovery plan, many extraneous factors must be considered since they impact, in some way, the extent or even the need for a plan.

Government Regulations

The Government can regulate the need and magnitude of a recovery plan by requiring certain data be available and accessible. In instances where government contracts are involved, how long data must be stored and what data must be maintained are dictated and adherence is mandatory. "Call it creeping government intrusion or just creeping government efficiency. Either way, federal and state agencies are increasingly issuing regulations that directly affect the operations of private-sector information systems." (Betts, 1989)

Independent Groups

In addition to the government, many independent groups can affect the viability of a given plan. A "hot site" developed prior to the asbestos scare could easily be located in a room with asbestos insulation, which would render it totally useless as a back-up location. In fact, any equipment located there would be subject to confiscation as being potentially contaminated. More directly related to computer centers are the recent discussions regarding new requirements on the insulation of wires. "The National Fire Protection Association

(NFPA) last week backed away from a plan to increase the fireproofing requirement on cables found in computer rooms as part of an expanded safety standard for data communications wires in building air spaces." (Booker, 1989) Had the requirements been changed, many recovery plans would have had to be reconsidered.

An often overlooked consideration of any alternate CPU solution is the software licensing agreement. Most of these agreements specify that the software will only run on one specific system. Frequently the software will not run on another CPU because the code is written to prevent it. It would be most unfortunate to have gone to all the trouble of developing a reciprocal agreement with another site only to find, in the middle of a crisis, that the applications software won't run on their system.

Costs Considerations

The greatest difficulty in developing an effective plan is the tremendous cost involved. Full "hot site" recovery plans can cost tens of thousands of dollars to maintain. Even partial "low cost" systems cost thousands of dollars a month to maintain. "By subscribing to CCSI and paying a monthly fee, users are buying 'a form of insurance' against such occurrences, J.R. Riconda, Vice President of CCSI, said. When smaller users are dubious over affording disaster recovery in the $10,000 to $15,000 price range, he is able to question whether they might not be interested in protecting the critical functions at a price closer to $6,000 to $7,000." (Mahnke, 1989) These costs do not include the development of the plan, any training or testing costs, or the cost of continually retraining and retesting the system as personnel changes occur.

PREVENTION BY DESIGN

Frequently a disaster can be prevented by proper design considerations. "The success and reliability of the data center often directly relate back to the details of the installation and integration of the various subsystems." (Aufenanger, 1989) Security, fire prevention and suppression systems, off-site storage facilities, uninterruptible power supplies, power conditioners and surge protectors, and environment control systems can reduce the likelihood of having a disaster or greatly reduce the impact on operations should one occur.

Power Interruptions

Since power fluctuations and interruptions are the most likely source of problems, surge protectors and UPS systems can reduce, if not eliminate, the probability of discontinued service from these sources. When choosing how extensive your UPS should be consider that, unless your operation is highly batch oriented, protecting the main system will not meet the need. All the terminals, workstations, file servers and multiplexers are not generally on the UPS. Also, if the power goes off most business operations cease anyway.

Fire and Fire Damage

Fire is another highly rated possibility. A well-planned fire suppression system that does not damage the equipment could reduce to an acceptable level the probability of damage due to fire. "What is a halon fire suppression system? Generally speaking, it is a system that automatically detects and suppresses fires that could occur from overheated electronic components or other Class A, B, or C materials." (Moore, 1989)

It is important to remember that a halon system only protects against the flame damage of a fire in the immediate area. Heat and smoke damage should also be considered when developing a prevention plan. Also, remember that in a real fire the fire fighters will have no knowledge of where the computers are so unlabeled locked doors will likely be opened with an ax and all your prevention literally goes up in smoke.

Flood

An important thing to remember in considering possible flood disasters is that most flooding does not occur because of rain. In fact most flood damage occurs from your own protection devices. The closest I have come to a real flood disaster was when the condensate pump for the air conditioner became unplugged. That pump was under the raised floor and had the interlock failed our own air conditioner would have generated a full fledged monsoon inside the computer room. Broken water pipes are another consideration, and remember an audible alarm is only good when someone can hear it.

Humidity

Depending on your definition of a disaster and how complex your environmental controls are humidity can cause serious problems. Most computer operators are aware of the dangers and warning signs of high humidity. Rain in the computer room or hail coming out of the air conditioner are fairly obvious problems. But what about low humidity? The byproduct of low humidity is static electricity and the signs are much more subtle. When the relative humidity drops to about 20 an astute manager might notice that the operators begin delivering reports to the most obnoxious of programmers (owing to the fact that the reports take on many of the properties of a charged capacitor). Most of the CPUs begin to have problems with an RH below 18. These problems can bring a system down when a simple humidifier, like you use when your kids get the croup, would have avoided the problem. Pans of hot water will also work in a pinch.

Personnel

Possibly the last asset to be considered in disaster recovery plan is personnel. In a real disaster the most devastating loss would be to key personnel. When developing a written plan remember that it is likely that the persons implementing the plan will be the least qualified to do so. Do not assume that the person restoring the back-ups knows how to do it, or where they are stored, or how to retrieve them, or even where the written plan is kept. By the way, copies of that written plan should be stored with your off-site back-ups. Otherwise your plan could be destroyed with your system.

Insure all MIS personnel are acquainted with the plan as well as several alternate personnel, just in case. Make sure they are cross trained so that the programmers as well as the operators know how to recover.

Keep written procedures for start-ups and restores that are simple enough that a novice computer user, like your boss's secretary, can use them.

An Once Of Prevention is worth A Pound Of Cure.

In many instances, pre-planning and design have contributed more to recovering from a disaster than an after-the-fact recovery plan. "Indeed, despite the best of intentions, significant investment, and mass quantities of documentation, most disaster recovery plans are likely to fail just when they are needed most....More often than not, luck plays as large a role in successful disaster recovery as skill and effort." (Rothstein, 1988)

> One of the most destructive hurricanes ever to pummel the U.S. proved to be largely impotent against the fortified data centers in the local [Charleston, SC] area.
>
> Flying in the face of damage estimates as high as $2 billion, local data centers contacted by Computerworld last week said the loss of power was the only blow Hugo managed to land to their operations.
>
> Data center managers credited preparedness, sturdy building design and providence for their successful defense in the Friday, Sept. 22 bout with Hugo. (Pastore, 1989)

AN ACTUAL SCENARIO

Since each case will vary considerably, let us consider the development of a cost effective disaster recovery plan for a particular site. MWSC is a medium sized public education institution with an enrollment of approximately 4,000. Currently three separate computer systems are maintained. The first is designated to handle the administrative functions of the college including payroll, registration, financial aid, student records, accounting, and scheduling. The second is dedicated to an on-line library system. The third is the student development system which is set aside for academic applications and student programming.

Access to all systems is controlled through a Micom data PABX system with all cabling terminating in a central computer room. The room is environmentally controlled and is physically located to minimize any possible natural disasters. It is on a hill, minimizing flood, on the first floor and partially underground, to minimize possible tornado damage.

Disaster Analysis

In this instance, an interruption is only considered a "disaster" if it results in loss of irreplaceable data, or if it causes discontinuance of service for a prolonged period (in excess of one week) or loss of service at a critical time (registration, for example). Since the system is primarily designed for on-line functions, any loss of power that affected the entire campus would not constitute a disaster since no one could use the system even if it were running. Similarly any thing which completely decimated the campus and/or the local community, such as a major earthquake or flood would not constitute a disaster because the need for the system would not be present.

Given these constraints, the most likely disasters are:

A power surge which destroyed all the computer systems (since the needs of the campus could be met to some degree even if two of the systems were destroyed) or destroyed the data PABX.

A fire that would destroy the computer room, or smoke, heat or water damage from a fire in an adjoining area.

A tornado destroying the computer room.

Additional Constraints

State and federal law require the maintenance of certain data. All such information is stored on historical tapes in the vault area. Redundant hard copies are maintained in a storage area in another building, and the majority have copies stored at the state capital. As no remote sites are currently in use, there is no need for concern as to their conformity to present standards.

Since MWSC is a public institution, considerable financial limitations exist. These restrictions eliminate the possibility of contracting with any service to provide back-up facilities and equipment, or the construction of an alternate computer room. In addition, there is no funding available for outside assistance in the development, testing, or implementation of a plan. Only the existing personnel and equipment are available for this project. Given the preceding conditions, can a cost effective disaster recovery plan be formulated for MWSC?

Disaster Reductions by Design

The possibility of a power surge destroying all the computer systems has been greatly reduced by the installation of a power conditioning and surge protection device. This equipment monitors and interrupts any power outside of prescribed conditions. The transformers and associated electrical equipment are physically located in an adjacent concrete block room to reduce the possibility of damage due to a lightning strike. This room is on the ground floor of a three story building making the probability of such a strike virtually nonexistent.

A complete halon fire suppression system has been installed, and is tested on a regular basis. The computer room itself is sealed to reduce the possibility of smoke or dust contamination, and it has its own air conditioning and humidification system. The structure of the room is reinforced concrete on two sides, and concrete block on the remaining sides. One side of the building is completely underground. This combination greatly reduces the possibility of fire and pollution from fire fighting efforts in adjoining areas.

In addition to the computer room, a separate attached vault is provided to accommodate magnetic tapes. The fire suppression and environmental controls extend to this room which is secured with a metal door. Magnetic tapes are also stored in a remote facility and are cycled to this room on a weekly basis. Access to the computer room is through metal security doors which automatically lock and are opened by a combination which is periodically changed and provided only to the computer room staff. The doors are inside the computer center which is secured by metal doors opened with a key. The distribution of the keys is controlled through campus security which also patrols the area to prevent unauthorized access.

Can A Cost Effective Plan Be Developed For MWSC?

MWSC is typical of many small to medium size businesses where financial limitations prevent the development of a fully comprehensive disaster recovery plan. The temptation is to dismiss the possibility that an effective plan can be developed. Most smaller organizations make the mistake of trying to fully emulate the larger institutions, and failing that, develop no plan whatsoever. The majority of potential disasters can be eliminated through low cost, procedure oriented methods. A procedure to back-up the data can be completely implemented, tested, and maintained at little or no additional cost. Materials to assist management in the development of such systems are readily available at minimal expense.

A plan for MWSC is neither prohibitively difficult or expensive. Maintaining such a plan requires continual re-thinking and updating, but that will be the case with any effective plan. Many of the preventative measures have already been designed in. If they had not, which is generally the case, the cost to benefit analysis would have to be performed for each acquisition but a plan can and should be developed.

FINAL CONSIDERATIONS

Selling The Plan

Unfortunately the very reliability of the equipment makes it difficult to "sell" disaster recovery. "If it's not broke don't fix it." There is a real temptation to "let something happen, just to teach them a lesson," don't circum it never works. There are a number of things that can be done without additional funding. Write up the best plan with what you have and work on acquiring the rest. Put something in writing, start-up, Back-up and Shut-Down procedures if nothing else. Do not let it slide. If the data processing people are not concerned about disaster recovery no one else will be either.

Another blockade is the size of the systems. I have often wished I could keep some of those old "boxes" as a shell just so I wouldn't have to explain why it costs so much for "that little thing." Its also difficult to explain why you want to have separate environmental controls for a small system. For those reasons I generally try to consolidate as much equipment as is feasible in one central computer room. The other side of that philosophy is that a localized disaster can destroy more of the system. But whichever philosophy you ascribe to do something.

It is a common misconception that an outside professional can design the best recovery plan for an organization. The on-site management team is much better prepared to analyze the fundamental needs of their organization than any extraneous group could ever become. In addition, the existing staff have no ulterior motives, such as a greater profit, in designing the system. They are in the best-all-around position to assess the validity of the plan and implement the necessary procedures to facilitate it. They also have more to lose if it is not done properly.

Given the constraints of a small to medium size company, it is unlikely that a plan to cover every possible situation can be developed. However a low cost plan could, and should be developed to reduce the possible effects of a disaster and allow the firm to continue operating at some reduced capacity. It is completely understandable that a manager would be unable to plan for every contingency. It is unconscionable that a competent manager would fail to plan for any.

References

Aufenanger, J. E. (1989, August). Protection by design, Service News, pp. 35 - 39.

Betts, M. (1989, April 24). Rules and regulations, ComputerWorld, p. 1, 115.

Booker, E. (1989, July 24). Fire group douses proposed standard, ComputerWorld, p. 8.

Buckland, J. A. (1982). EDP Disaster Recovery Planning and Procedures, I, 2-2.

Business Recover Systems, Inc. (1986). Practical Guide to Disaster Recovery Planning.

Harris, A. (1989, May). Planning your recovery, Service News, pp. 22 - 36.

Harrison, B. (1988, September 9). Planning for disaster, TPT, p. 23 - 30.

Hawley, N. (1989, March). En Garde!, Service News, p. 26 - 37.

King, P. (1989, April). Disaster Recovery Options, HP Professional, pp. 36 - 41.

Leibs, S. (1989, April 17). Arresting Horror Stories, Information Week, pp. 30 - 32.

Leverentz, J. R. (1988, May). Disaster Planning, HP Professional, pp. 34 - 36.

Magee, D. (1987, October). Options For Disaster Recovery, Supergroup Association Magazine, pp. 20 - 30.

Mahnke, J. (1989, January 23). No-Frills Disaster Service Options, Information Systems, p. 11.

Michel, L. (1989, August). Disaster planning and recovery: A true story, Interact, pp. 95 - 104.

Moore, J. F. (1989, May). Halon to the rescue, Service News, p. 37 - 40.

Rothstein, P. J. (1988, October 15). Up and running: how to ensure disaster recovery, Datamation, p. 86 - 96.

Nielsen, A. C. (1986, September 9). Ready for disaster? Computer Decisions, pp. 64 - 78.

Pastore, R. (1989, October 2). Data centers defy Hugo punch, ComputerWorld, p. 8.

Robinson, L. (1989, September). Disaster recovery, protecting your business, Interact, pp. 104 - 107.

Ryan, A. J. (1989, May 15). Cosmic swap no problem for Atlantis, ComputerWorld, p. 4.

Scheier, R. L. (1989, June 26). For many, backing up is hard to do - right, PC Week, p. 69.

Schlandweiler, J. (1989, September). The strategic value of network disaster recovery planning, Networking Management, pp. 54 - 60.

Severinsky, A. (1989, August). Warning: Power may be hazardous to your system, Service News, pp. 24 - 33.

3803

HARDWARE SUPPORT CONTRACTS

Margaret Brunner
Northern California Cancer Center
32960 Alvarado-Niles Road, Suite 600
Union City, California, 94587-3106
(510) 429-2539

OVERVIEW

This paper addresses the issue of hardware support contracts in a data processing shop. I will address how to find alternative support vendors as well as how to evaluate the options for the systems you want covered by a hardware support contract. I will conclude with charts comparing the prices among 5 vendors from the San Francisco bay area. This will be presented in the context of my installation which includes two HPS - a 3000/52 and a 9000/845s.

INTRODUCTION

I have been using HP3000s since 1978. All the organizations that I have been associated with have always carried HP contracts. HP support has been excellent. Unfortunately last year, due to loss of revenue, I was asked to review the maintenance cost of all equipment and come up with a plan to contain or to lower the cost. This request resulted in the investigation of alternatives that produced the data for this paper.

To give you some background, we had at that time on the HP9000/845s 40 megabytes of main memory, 18 ports and 2.68 gigabytes of disc space. Its peripherals included a 600 lpm dot matrix printer, a 1600/6250 bpi tape drive, and two modems to allow off site communication with the computer.

This system is principally used by our statisticians and epidemiologists.

We also had at that time on the HP3000/52 8 megabytes of main memory, 32 ports and 2.95 gigabytes of disc space. Its peripherals included a 1200 lpm dot matrix printer, a 1600/6250 bpi tape drive with data compression capabilities, a modem, and a statistical multiplexor with 4 ports.

This system is principally used by our data entry personnel and data processing personnel.

Both these computers are linked across a local area network with 26 PCs.

Our organization gathers cancer tumor data from hospitals for research purposes. The nature of our business means that the computer is central to the operation of the organization but if the system goes down, next day repair service is acceptable. We are non profit and exist from grants and contracts, therefore, saving money is more critical than downtime.

Prior to even starting an investigation of alternatives, I would analyze the needs of the organization and the nature of the business. Have an idea of what is absolutely necessary to the organization versus what would just be nice to have. Consider whether a piece of equipment must be running and available 7 days a week, 24 hours a day or if the equipment can be down 1 to 2 days or if the equipment is non critical and can be removed from the office for a period of time. Evaluate all your equipment in these terms.

SUPPORT CONTRACT VENDORS

In order to save money on contracts, you must first investigate whether there are any alternative vendors available in your area.
Support contract vendors are area specific for the most part. In order to find the vendors in your area, either attend a Regional Users Group meeting and speak with the vendors, or attend an Interex conference and speak with the vendors, or read the advertisements/classifieds in publications such as Interexpress, HP Chronicle, Interact, and SuperGroup Magazine. What you need to look for in the ads is the key word "repair" and/or "maintenance".
You can also just ask other managers if there are any alternatives available in your area.

Once you have found the vendor names, contact them and let them know that you are in the market for support.

Make a complete list of the hardware equipment you want on contract or, alternatively, use the current contract you have. Send (or fax) this list to the vendors you contacted, requesting bids.

Because this paper is more about methodology than findings, I decided not to publish the vendor names that I contacted. In the charts at the end of the paper that I created to compare costs among vendors, the vendors are simply labeled as A, B, C, D, and E.

SUPPORT CONTRACT QUESTIONS TO ASK

This following section addresses questions to ask of the vendor and of their references. I listed many questions. You can pick and choose among them depending on your specific needs and level of comfort in asking the questions. As an added task, besides asking questions, I would highly recommend visiting the vendor's facilities and seeing their setup.

QUESTIONS TO ASK OF THE VENDOR

QUESTIONS TO ADDRESS THE VENDOR'S RELIABILITY

- How large is your corporation?
- What distinguishes you from the competition?
- How long have you been in business?
- How many sites are you currently supporting?
- How many systems are you currently supporting?
- How many customer engineers (CE) do you have on staff?
- What are the features you pride yourself about?
- Do you have accounts my size and/or with my equipment? Who are those accounts?

What you are looking to answer with these questions is the likelihood of the vendor remaining in business. This is of particular importance if you are prepaying for their service. What you are also interested in is if they have the staff, and the experience to support your installation.

QUESTIONS TO ASSURE RECEIVING THE MOST ADVANTAGEOUS PRICE

- What is the price differential between all your support options?

- Do you provide any other type of discount for pre-payment, etc.?
- Does support need to be prepaid or will I be invoiced?
- Is the invoice monthly, quarterly, or yearly?
- What are the payment plans available?
- What are other discount opportunities?
- Do you offer any discounts in other areas if I will carry a hardware contract with you?
- Are you offering any bonus if I sign on with you?

I usually let vendors know that I am looking at various vendors. I find that this helps produce the most information and the best price and conditions. There are also many types of discount possibilities. Some offer discounts for pre-payment, some offer discounts for 1 to 3 year contracts, some offer price freezes for up to two years, some offer services or discounts for signing with them. Investigate all possibilities to negotiate the best contract.

QUESTIONS CONCERNING PART REPLACEMENTS

- Where do you get parts from? Where is your warehouse?
- How much stock do you have on hand within a couple of hours driving distance? Where are the parts to my system warehoused?
- Do you have my hardware (specific parts) in stock?
- Will you cover all my equipment (including non HP)?
- Are complete units in stock?

These questions concern the amount of downtime you might experience if the system fails. If parts are not kept in stock locally and have to be ordered from HP or mailed from other warehouses in the state or the country, then this will add to the downtime.

QUESTIONS CONCERNING SERVICES PROVIDED

- What are your hours of coverage?
- How many site visits/preventative maintenance visits are included in the price?
- Do you do installs, deinstalls and moves? Would this be part of the contract or extra?
- What is your usual response time versus your

guaranteed response time?

- Does your price cover all parts, labor and travel time?
- Is there an extra charge because of where you are located in relation to where I am located?
- Do you offer "predictive" software support?
- How many account reviews are included in the price?
- Do you do depot repair of PCs or printers(workstation products)?
- Can you rent or lease equipment or do you carry loaner equipment?
- Do you do performance consulting?
- Do you sell refurbished equipment?
- Do you work to completion or work to within contract hours?
- Do you do engineering improvement modifications as part of your standard contract?
- Do you provide remote support through remote diagnostics to determine the cause of problems?
- Do you have after hours telephone coverage?
- Do you have a maintenance site log program?
- Do you provide environmental site surveys?
- Do you have a central dispatch system? What does it consist of?
- Will I receive a written report after every service call?
- Will you support non-HP peripherals or components?

Make sure you thoroughly compare services that are of interest to you because vendors vary all over the board in what they offer. Some offer monthly account reviews, some quarterly, some semiannually, and some annually. Some will offer quarterly preventative maintenance (PM) while some do it semiannually. Some will throw in free services for you such as performance evaluation, free hardware product trials, or free deinstalls, installs, and relocation if you are their customer while some will charge for the same service or not provide it at all. Some will provide other services at a discounted rate if you carry a hardware contract with them. Some will allow you to call in HP at their cost if you are dissatisfied with their service or will pay whatever HP charges to reinstate you as an HP support customer if you are unhappy and decide to return to HP. It pays to investigate and evaluate all these services.

QUESTIONS CONCERNING THE SERVICE PERSONNEL

- Does the CE have background and training in the operating system?
- Are the CEs former HP personnel, homegrown or HP factory trained on my specific equipment?
- What is the name and background of the specific CE assigned to my site?
- Is the assigned CE trained on my machine and operating system?
- How many CEs are on staff and fully trained? How will they be trained to match my needs?
- How do your CEs stay abreast of currently released products and original equipment manufacturer (OEM) engineering improvements?

These questions are of vital concern because basically you are paying for the CE's knowledge and availability. How sharp that person is and how well trained they are will have a direct impact on how long you will be down. Also the number of CEs on staff versus the number of systems maintained is critical because you do not want all the CEs out on call if you have a hot problem. Make sure the CE assigned to you knows the operating system commands in order to perform both online and offline diagnostics. This is especially critical if you have a UNIX system.

PROCEDURAL INFORMATION

- What is the procedure for contacting the CE?
- What is your escalation management process?
- What is the after hours procedure to follow?

This is important especially if you have a very hot problem. You need to know what resources will be brought to bear on it, who will pay for these resources, and the time frame involved before the big guns are brought in.

QUESTIONS TO ASK REFERENCES

- How does the vendor compare to HP in terms of support?
- What is your level of satisfaction with this vendor?
- What is the nature of the problems that you have had with your system?
- How long have you had a contract with this vendor?

- Is there anything that really stands out in terms of pros or cons?
- What type of contract do you carry with this vendor?
- Did the vendor deliver the service as contracted (ie account reviews, environmental site reviews, PMs, written reports, remote diagnostics, OEM engineering improvements, etc.)?
- Did you look at other support vendors?
- Why did you choose this vendor?
- Who else did you contract with in the past? Why did you drop that vendor?

Be sure to ask the vendor for current references with setups similar to your own. Be sure to spend the time and talk with the references given. References will provide you with invaluable information.

ITEMS OF CONCERN

The following paragraphs detail some items that caught my attention and might not be real obvious to everyone.

Check out all statements made by the company because they are not always accurate. Either speak with their references or call HP depending on the nature of the statement. For example, Company B states in their brochure "...regularly schedules both on-the-job and off-site training sessions to either cross-train in currently released equipment or to become proficient in newly released products. For newly released equipment we utilize classes provided by the OEM." HP on the other hand told me they will provide training courses on older equipment but not on newly released equipment such as the newer disc drives that I have or the 1200 lpm printer that I have or the HP9000/845. HP will also provide Hardware Support Subscription Services to anyone on older equipment such as 2564 line printers, the 7980 tape drives, the series 52, etc.; but not on newer pieces of equipment such as the 2566 line printer, the series 845, or the series 6000 disc drives. If improvements come along on these units, HP does not offer them outside.

Be real specific on inventory levels, especially of newer equipment. Vendors can provide support at a real competitive price because they can acquire older parts and equipment inexpensively. It is far more expensive to stock parts for newer equipment. For example, the

processor board on the 845 costs around $50,000. When you visit the vendor's facilities, ask to see the specific parts in stock that you are interested in, if this is of concern to you.
Some software services are only available if you carry a hardware contract with that vendor. For example, HP will offer NETASSURE only if you carry the hardware contract for that machine with them. Be as concerned with what you will not receive from the vendor that you want or currently have as well as what you will receive from the vendor.

Double check the quotes coming back to you because the vendors make mistakes on quantity and part numbers. They might also inadvertently drop a part from the contract that you want on contract. All these errors affect the total and can lead to erroneous comparisons.

SUPPORT CONTRACT COSTS

The 4 charts at the end of the paper detail the system components for both the 52 and the 845 and show the price differentials among the 5 companies I looked at within the San Francisco bay area. As you will be able to see there is a great deal of difference among the companies. This applies not only to the total amount but also applies at the component level. Rankings among vendors also shift depending whether you are looking at a full service support contract or a next day service support contract.

In the charts, I decided only to show two of the alternatives available which are Full Service Support and Next Day Service Support. Full Service Support is Monday through Friday 8:00 am through 9:00 pm coverage with a 4 hour response time except for company D whose hours are Monday through Friday 8 am through 5 pm. Next Day Service Support is Monday through Friday 8:00 am through 5:00 pm coverage. There are other types of coverage available such as 24 hour, 7 days a week coverage.

For my purposes, less extensive coverage was fine because we do not run a 24 hour 7 days a week shop and we can afford downtime on our computers. Please note that if you go with next day service contract, you can if necessary have same day service if you request it on an as needed basis for a flat uplift charge.

Prices shown for vendors B through E on the charts are based on quarterly prepaid support. But the prices shown do not tell the whole picture. There are many other discount options available to reduce the price. Each vendor has his own flavor. The following is a list of some of the options available depending on the vendor:

- .10% less for a 12 month pre-paid contract
- .1 month free for a 12 month pre-paid contract
- .5% less for a 3 year contract
- .Current price frozen for 2 years
- .1 month free for signing on with them
- .3 month to 5 year contracts are also available

CONCLUSION

There are a broad range of prices and a variety of options and ways to save. Be careful in your study and comparison of vendors and consider all services offered versus only the cost of the package. It might be unwise to go with the cheapest, if the service offered is also minimal. Go for the best price offered for the type of service you desire. This might not be the cheapest, but should be the one that will offer you the most overall for the least amount.

One last thing to keep in mind is that you can split systems up among different vendors if the cost and service warrants it, especially if different systems have different needs (i.e., one is a production system and the other a development system or, in my case, one is a UNIX system while the other is an MPE system.) You do not need to place all your systems with one vendor or you can just place one system with a vendor to test out their service before moving all your systems over.

Chart 1 - SAME DAY SUPPORT FOR THE 52

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES
FOR THE HP3000/52 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
2	GIC 30079A	32.00	16.00	15.00	22.00	10.00
1	S40 Fron 30170K	18.00	12.00	9.00	9.00	0.00
1	S40 Power 30170P	14.00	7.00	3.75	5.00	0.00
1	58 Mem Cntrl 30172A	0.00	0.00	7.50	8.00	0.00
1	Basic Module 30477B	217.00	168.00	89.25	105.00	100.00
2	4MB memory 30479A	0.00	0.00	0.00	0.00	0.00
1	Power Supply 63909F	23.00	15.00	11.25	14.00	0.00
4	ADCC-MAIN 30018A	76.00	20.00	27.00	36.00	16.00
4	ADCC-EXTENDER 30019A	76.00	20.00	27.00	36.00	16.00
3	571MB Disk 7937H	165.00	120.00	81.00	114.00	90.00
1	571MB Disk 3937XP	55.00	42.00	28.00	39.00	35.00
1	670MB HPIB Disk C2203A	25.00	20.00	18.00	18.00	30.00
1	Dot Matrix ptr 2934A	31.00	12.00	18.00	15.00	25.00
1	1200 lpm pt 2566C	237.00	190.00	146.25	178.00	125.00
1	6250/1600 tape 7980XC	57.00	35.00	45.00	33.00	40.00
1	Thinlan link 30240A	0.00	3.00	3.00	2.00	4.00
2	X.25 Statmux 2335A	82.00	40.00	30.00	30.00	40.00
2	Codex 2640 Modm 32066A	68.00	56.00	4.00	30.00	40.00
	TOTAL	1176.00	776.00	563.00	694.00	571.00

Chart 2 - SAME DAY SUPPORT FOR THE 845

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES
FOR THE HP9000/845 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
1	HP9000/845 SPU A160	383.00	293.76	240.00	300.00	50.00
3	16MB ECC RAM A1037A	0.00	0.00	0.00	0.00	0.00
1	HP-CIO HP-IB 27110B	4.00	2.88	3.00	2.00	4.00
1	S800 CIO 27111A	5.00	3.84	3.00	3.00	5.00
1	6 Channel MUX 27140A	7.00	9.60	4.00	4.00	0.00
2	Async CIO Mux 98196A	14.00	7.68	12.00	10.00	14.00
1	Backup Unit A1014M	29.00	21.50	13.00	17.00	20.00
1	LAN 9000 S800 91786B	9.00	4.80	15.00	7.00	5.00
1	THINMAU 28641A	4.00	1.92	5.00	5.00	3.00
1	6250/1600 tape 7980A	57.00	33.60	43.50	44.00	35.00
2	1.34mb FL Disk C2204A	74.00	55.68	54.00	54.00	60.00
1	HP 700/92 Term C1001G	7.00	3.84	6.00	4.00	5.00
1	600 LPM printer 2564B	112.00	72.00	56.25	60.00	60.00
1	8-pen Plotter 7550A	40.00	19.20	24.00	24.00	20.00
1	LaserjetIII 33449A	40.00	32.64	28.00	29.00	20.00
	TOTAL	785.00	562.94	506.75	563.00	501.00

Chart 3 - NEXT DAY SUPPORT FOR THE 52

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES
FOR THE HP3000/52 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
2	GIC 30079A	28.00	12.80	13.50	19.80	8.00
1	S40 Fron 30170K	9.00	9.60	8.10	8.10	0.00
1	S40 Power 30170P	7.00	5.60	3.38	4.50	0.00
1	58 Mem Cntrl 30172A	0.00	0.00	6.75	7.20	0.00
1	Basic Module 30477B	174.00	134.40	80.33	94.50	80.00
2	4MB memory 30479A	0.00	0.00	0.00	0.00	0.00
1	Power Supply 63909F	19.00	12.00	10.13	12.60	0.00
4	ADCC-MAIN 30018A	60.00	16.00	24.30	32.40	12.80
4	ADCC-EXTENDER 30019A	60.00	16.00	24.30	32.40	12.80
3	571MB Disk 7937H	129.00	96.00	72.90	102.60	72.00
1	571MB Disk 3937XP	43.00	33.60	25.20	35.10	28.00
1	670MB HPIB Disk C2203A	20.00	16.00	16.20	16.20	24.00
1	Dot Matrix ptr 2934A	16.00	9.60	16.20	13.50	20.00
1	1200 lpm pt 2566C	190.00	152.00	131.65	160.20	100.00
1	6250/1600 tape 7980XC	46.00	28.00	40.50	29.70	32.00
1	Thinlan link 30240A	0.00	2.40	2.70	1.80	3.20
2	X.25 Statmux 2335A	46.00	32.00	27.00	27.00	32.00
2	Codex 2640 Modm 32066A	58.00	44.80	3.60	27.00	32.00
	TOTAL	905.00	620.80	506.74	624.60	456.80

Chart 4 - NEXT DAY SUPPORT FOR THE 845

MAINTENANCE COST COMPARISON AMONG 5 COMPANIES
FOR THE HP9000/845 HARDWARE

QTY	HARDWARE	EXTENDED PRICE FOR MAINTENANCE				
		A	B	C	D	E
1	HP9000/845 SPU A160	306.00	244.80	216.00	270.00	200.00
3	16MB ECC RAM A1037A	0.00	0.00	0.00	0.00	0.00
1	HP-CIO HP-IB 27110B	3.00	2.40	2.70	1.80	3.20
1	S800 CIO 27111A	4.00	3.20	2.70	2.70	4.00
1	6 Channel MUX 27140A	5.00	8.00	3.60	3.60	0.00
2	Async CIO Mux 98196A	10.00	6.40	10.80	9.00	11.20
1	Backup Unit A1014M	23.00	17.92	11.70	15.30	16.00
1	LAN 9000 S800 91786B	7.00	4.00	13.50	6.30	0.00
1	THINMAU 28641A	2.00	1.60	4.50	4.50	2.40
1	6250/1600 tape 7980A	46.00	28.00	39.15	39.60	28.00
2	1.34mb FL Disk C2204A	60.00	46.40	48.60	48.60	48.00
1	HP 700/92 Term C1001G	4.00	3.20	5.40	3.60	4.00
1	600 LPM printer 2564B	90.00	60.00	50.63	54.00	48.00
1	8-pen Plotter 7550A	23.00	16.00	21.60	21.60	16.00
1	LaserjetIII 33449A	22.00	27.20	25.20	26.10	20.00
	TOTAL	605.00	469.12	456.08	506.70	400.80

Paper # 3804
The Care and Feeding of CASE Tools

Stephen L. Day
Ernst & Young
One James Center
901 East Cary Street
Richmond, Virginia 23219
(804) 775-4433

So! You've decided to buy a CASE tool. There are many happy owners of CASE tools in the world, with their tools giving them many benefits for their investment. But, if you should choose a CASE tool that is not right for you, then both you and the CASE tool will be very unhappy.

What are the qualities that a prospective CASE tool owner should look for in a CASE tool? The three most important aspects are the environment the CASE tool will live in, whether the CASE tool will be used for many various duties or only a few, matching the duties expected of the CASE tool with the tool's strengths, and being ready to give the CASE tool the appropriate food, training, and veterinary care it requires.

Acquisition

One of the most important decisions you will make as an MIS Manager is the decision to acquire a CASE tool. Once this decision has been made, the choice of "which one?" becomes critical. Aspects such as the ability of your organization to accept the changes associated with introducing a CASE tool, the functions you want the tool to perform, and the appropriate resources to bring to bear in preparation for implementation are important factors in your decision.

Organizational Assessment

The first step in the acquisition of the new CASE tool within the organization is to assess the current position of readiness for the organization to accept the tool. When a new tool is introduced to the organization, it can have a big impact on the organization's life style. There will be people who were very happy with the way things were before the CASE tool showed up. This can be most acute in small children or older members. They will attempt to ignore the new acquisition, and perhaps abuse it when no one is looking.

Other organization members will look at the new acquisition as a lot of fun to play with. They will be very eager to have fun with the new tool, but will not be willing to put in the time and effort required to put the tool to productive work.

There will exist within the organization people who are willing to make changes to accommodate the new tool, to undergo the necessary training to make it work, and to adhere to the rigor required to effectively use the tool productively. These people should be identified early in the tool acquisition process, and the tool should be placed with these people initially so that it will have the opportunity to "get off to a good start."

One of the best ways to identify these nurturing people is to do an "Organizational Assessment" as one of the first steps in the acquisition process. One of the facets of the assessment, and we will talk about others, is to measure the organization's ability to accommodate change. The people who are identified as natural leaders, innovative workers, and willing to change are selected as candidates for the first contact with the CASE tool.

Another important aspect of selecting a CASE tool is what you will ask the CASE tool to do. Some CASE tools are very powerful, but require a lot of maintenance, require very powerful and expensive "houses (workstations)," and are targeted at different environments. Others are general purpose, and might be targeted at a specific task or narrow range of tasks. Still others are very pretty to look at and entertaining to play with, but not very functional.

The Organizational Assessment should identify the tasks which the CASE tool will be expected to perform. The requirements are based on the business objectives of the organization.

Technology Assessment

Once the objectives for your new acquisition have been developed, it is time to learn about the different "breeds" of tools available on the market. How closely you match the capabilities of your new tool to the abilities of your organization and the objectives of implementing the tool to the tool's strengths will largely determine your satisfaction with the tool.

The extent of your search will be bounded by the environment which you will be able to provide for your tool. The hardware environment your CASE tool will operate in, the target environment your tool will be developing systems for, and the methodology and techniques your tool will be support are extremely important factors to consider in setting the scope of your investigation.

For most people, the proper CASE tool will be a function of the methodology employed in the development of the systems. If the methodology employed is a very loose set of tasks defined by the individual project manager, the most flexible tool available may be the correct choice. If a formal methodology has been adopted by the organization, the techniques and deliverables called for in the methodology will guide the selection toward a tool that supports these same techniques and deliverables.

Another aspect of the tool choice is code generation. There are several tools available which may assist in the business analysis or technical design of systems which poorly support code generation. Some tools may support code generation, but not build on the products of other tools. The code generation issue will greatly impact your choice of tools until a Developer's Workbench including code generation becomes a reality.

Methods Assessment

The adoption of a CASE tool can have a profound impact on the life style of the organization. Just as the implementation of a system within a user department will change the way the user department operates, so too will the CASE tool change the way the IS organization operates.

With the adoption of the tool, you may want to ask yourself whether or not you want to automate the existing manual processes, or whether you want to make improvements in the process of developing software. Since the tool supports the process, and not the other way around, if the methodology used to develop systems is going to change, then it makes sense to decide how you will build systems before buying the tools.

Evaluation

Once you have a set of objectives for your selection of a CASE tool, it's time to go shopping. When you begin visiting CASE vendors, you will find that the prices are almost as varied as the functionality of the tool. Bear in mind also that the cost of the tool itself is a minor part of the acquisition cost. Training in the use of the tool, new techniques, coaching in the use of the tool, and in learning curve will expand the cost of the tool to 2-5 times the cost of the software.

Several questions you may want to ask your prospective tool vendor are:

> *Where does the tool store metadata?* Most CASE tools have a dictionary or encyclopedia which holds the information collected through the CASE tool. Sometimes, this information is shared among different users of a tool or is of an open architecture to allow sharing among dissimilar tools. The more flexible and open the metadata model, the more sharing of information you will have among the development staff.
>
> *Is the tool integrated throughout the development life cycle?* Tools labelled I-CASE (Integrated CASE) have a common metadata model that shares information from Strategic Planning through Construction. Others may be targeted only at logical database design or screen generation. The degree to which the tool spans the entire life cycle or allows sharing throughout the life cycle will determine the amount of development leverage you will achieve from Planning to Analysis to Design to Construction.

Does the tool support multiple developers sharing a common metadata model? On systems of any larger size, duties and responsibilities for system components will be distributed across an application team. Management of this effort is complicated if the tool does not enhance this division of effort.

Does the tool work with or enhance the management process of a project? Project management is an essential part of any software development effort, with or without CASE tools. Through the definition of deliverables, project management can be enhanced through the automatic recognition of deliverable completion or the measurement of compliance with business objectives.

Does the tool enable model-based development and maintenance? Many tools will generate skeleton programs which contain some percentage of a completed application. Once these skeletons are enhanced for full functionality, however, regeneration of the application will eliminate the enhancements. If the models are not able to be used for maintenance, then the models become out-of-date when compared to the program code.

Does the tool generate applications for multiple hardware platforms? Most importantly, does it generate applications for all the hardware platforms that exist in your technology architecture?

Deployment

Once the decision has been made on the appropriate CASE tool for the organization, it is time to start creating an environment where the tool will be happy within the organization. A CASE tool, like most pets, will require a period of adaptation, during which the organization may feel some discomfort.

A first step in introducing the CASE tool into your environment is to gain experience through the use of a Pathfinder Project. This project is a controlled, coached project which utilizes the tool on a small, non-critical system. The training for the project team is performed on a just-in-time basis, and the team members are used to seed second-generation projects after the completion of the pathfinder.

An infrastructure of experts or "Masters" can also be established as projects are completed. These people are identified as having the skills to coach and mentor others in the use of the tools, as well as having tool expertise.

As your CASE tool grows up, it will become more and more a part of your development organization. As your organization gains experience in using it on projects, you will adapt the standard methods and techniques for developing deliverables to what your project experience shows to be "best practices." You may even select someone to be a "Chief Methodologist" who has responsibility for monitoring the methodology, changing it as necessary, and working with your CASE tool vendor to help the tool grow into what you want it to be.

Implementation

Hopefully, your acquisition of a CASE tool will yield many hours of fruitful systems development for your organization. If you give it the proper care, such as regular feeding in the form of training and coaching, occasional visits to a professional who can help you with suggestions on improving your CASE tool's performance, and nurture it by providing it with good people who have the abilities to use it, your experiences with your CASE tool should be happy ones.

INTEREX Computing Management Symposium Paper 3806

The original ALLBASE idea: IMAGE and SQL

F. Alfredo Rego

Adager

Sun Valley, Idaho
83353-0030
U.S.A.

Historical background

In early 1986, I met with several Hewlett-Packard executives in Cupertino who were very excited about the name that they had devised for their new strategic database product, ALLBASE. "The beauty of this concept," they said, "is that it will integrate the best practical aspects of IMAGE with the best theoretical aspects of the relational model for database management." They asked me if I liked the name and the concept behind it. I said that both the name and the concept were wonderful. I still think so now, after more than five years.

A recent conversation I had about the relational model for database management with its creator, Dr. Edgar F. Codd, makes HP's initial conception of ALLBASE even more relevant.

My conversation with Dr. Edgar F. Codd

At the SCRUG meeting in Pasadena, on May 9, 1991, I interviewed Codd in the setting of a public forum. I began my conversation with Codd by explaining that I had selected the session's title, "Understanding Databases," because it was ambiguous. We can choose to interpret the word "understanding" either as a verb or as an adjective. As a verb, "understanding" means that we are taking some action to try to understand what databases are. As an adjective, "understanding" means that we are talking about databases that treat us in a motherly fashion, that are tolerant, compassionate and sympathetic, that never break down, that always perform beyond the call of duty, that don't require expensive maintenance.

I then asked Codd if he knew of any such magically understanding databases. Laughing, he said, "Not at all." Addressing the audience, I asked if anybody else knew of any such magically understanding databases. More laughter. Funny, I thought, after having waded through all the glossy literature. Given this reality, we decided to interpret the word "understanding" as a verb. I invited Codd to help all of us in our efforts to try to understand what databases are.

As the basis for our discussion, I selected some key ideas from Codd's recent book (*The Relational Model for Database Management, Version 2*, Addison-Wesley, 1990).

The evidence would suggest that few people understand the relational model for database management (although many people certainly know SQL very well). I encouraged everyone to read and to study Codd's book because it brings together, under one cover, his fundamental ideas. To illustrate, I quote from the book:

"Four important points concerning relations follow:

1. every relation is a set;

2. not every set is a relation;

3. every relation can be perceived as a table;

4. not every table is a correct perception of a relation.

Designers of the relational DBMS products of many vendors appear to be ignorant of these facts or to have ignored them" (page 27).

"Of course, in many of the relational DBMS products on the market today, support for the integrity features of the relational model is quite weak. This weakness reflects irresponsibility on the part of DBMS vendors" (page 435).

Given these facts, I asked Codd about his feelings whenever people use "SQL" as synonymous with "the relational model for database management." Codd proceeded to clarify the myths surrounding SQL, with particular attention to the overselling of SQL as the standard for relational database management systems. "SQL is just a data sublanguage invented in late 1972 by a group in IBM Research, Yorktown Heights, NY. Although it was claimed that the language was based on several of my early papers on RM, it is quite weak in its fidelity to the model," Codd said.

"How was SQL ever adopted as an ANSI and ISO standard?", I asked. Codd replied, "That's an excellent question; I wish I knew the answer." (For those interested in pursuing this issue, Codd has devoted chapter 23 in his book to discussing the serious flaws in SQL.)

Because Codd had dedicated his book "To fellow pilots and aircrew in the Royal Air Force during World War II," I knew that he would, as a pilot, appreciate the fact that airplanes are amazing things that come in all shapes and prices. There is one airplane that went around the world without refueling; it was very slow, extremely uncomfortable, as fragile as a kiss and, therefore, unable to go through the storms and the turbulence that commercial jets usually encounter. There are huge, slow cargo planes. There is the Concorde. There are business jets. There is Air Force One, with a bed and (we would assume) a shower.

When I asked Codd about his recommendation for the "standard airplane" that everybody must have, he said, "Such a thing does not exist." While on this topic, he referred to page 22 of his book: "I believe that the days of monstrous programming languages are numbered, and that the future lies with specialized sublanguages that can inter-communicate with one another." To me, this sounded very similar to what is expected of open systems, whose existence depends on specialized things that inter-communicate well, as opposed to monstrous things that try to be all things to all people.

Codd shared with us his observations about the main shortcomings of the wishful implementations of the relational model, using IBM's DB2 to illustrate. "DB2, with about 50% compliance with the relational model, is the most faithful implementation," Codd said. "But even DB2 is still a long shot. The main problem is that all so-called relational database management systems do not support the fundamental features of the relational model. The fact that they may (or may not) support some other features does not relieve them of the responsibility of supporting the fundamental features. Without sound fundamentals, any structure is bound to collapse eventually. Everywhere, users are losing their patience. Things are taking too long and are too expensive. There is a deluge of marketing hype," Codd complained.

Regarding my question about any hope for the convergence of these so-called relational database management systems towards the relational model, Codd explained that, "Due to some fundamental decisions that the implementors had made early on in the game, it would be very difficult for them to converge towards the relational model."

I then brought up an issue that is highly relevant to the members of the HP3000 community who have developed high-quality, reliable applications based on IMAGE. Why should these people migrate to a poor-quality, unreliable, expensive and non-compliant so-called relational DBMS? Codd had only a couple of minutes to address this question, as it was the last question before lunch. Codd quickly mentioned that "Any conversion is a very expensive proposition in terms of labor costs, since automatic conversions are ineffective and need a great deal of babysitting." Codd did not foresee labor costs decreasing. "Therefore," he reasoned, "even though it would be an expensive migration, everyone should convert to a so-called relational implementation as soon as possible." En route to lunch, an IMAGE user approached us noting that the word "possible" might best be interpreted to mean "economically feasible."

There is no question in my mind about the ever-rising costs of conversion, but I do not agree with conversion for conversion's sake. I believe that there is only one valid reason for converting: to escape from a poor database management system that is not able to support vital applications. And then, people should only convert to a clearly outstanding database management system. Anything else is an exercise in futility.

REFERENCES

Brown,A.J.,"Distributed Data Management From HP", Proceedings INTEREX HP Users Conference, San Francisco, CA, September 11-14, 1989.

Connors, T. and Lyngbaek, P., "Providing Uniform Access to Heterogeneous Information Bases", Software Technology Laboratory, Report STL-87-16, Hewlett-Packard Laboratories, December 4, 1987.

Date, C.J., "Distributed Database Technology", Proceedings DB/EXPO'89, San Francisco, CA, January 31 - February 2, 1989.

Eunice, J., "Leaders in Object Databases", D.H. Brown Associates, Inc., December 1988, 400 Benedict Avenue, Tarrytown, NY, 10591, (914)631-7859.

Fishman, D.H., Beech, D., Cate, H.P., Chow, E.C., Connors, T., Davis, J.W., Derrett, N., Hoch, C.J., Kent, W., Lyngbaek, P., Mahbod, B., Neimat, M.A., Ryan, T.A., Shan, M.C., "Iris: An Object-Oriented Database Management System", ACM Transactions on Office Information Systems, 5(1), Jan 87.

Larson, O.J., "Strategic Importance of Relational Database Technology", Proceedings INTEREX Computing Management Symposium '89, Nashville, TN, March 8-10, 1989.

Perreault, R., Rotzell, K., "Object Database Management Systems", INTERACT, April 1990, pp.114-126.

Schussel, G, "Database Developments in the 5th Generation", Proceedings 1989 Database and Cooperative Processing Symposium, Washington, DC, March 20-22, 1989.

Stein, J., and Maier, D., "Concepts in Object-oriented Data Management", Database Programming and Design, 1(4), April 1988, pp.58-67.

Tash, J., "Application Architecture and the Strategic Role of Relational DBMS", Database Decisions, Inc., 41 Marcellus Drive, Newton, MA, 02159, (617)332-3101.

White, C., "The Importance of SQL", Proceedings DB/EXPO'89, San Francisco, CA, January 31-February 2, 1989.

White, C., "Emerging Technologies for The 90's", Proceedings DB/Expo'91, San Francisco, CA, March 25-28, 1991.

databases. Right now, HP is seriously considering the obvious evolution of ALLBASE/Turbo CONNECT: SQL read *and* write access to IMAGE databases.

How serious is HP about implementing SQL read/write access to IMAGE databases? HP is very serious, indeed, but it needs *your* input to help define the future directions for its database programs. On this topic, *The HP Chronicle* (on page 20, May 1991 issue) has an article, "HP seeks customer input on databases", quoting Doug Dedo, HP IMAGE product line manager. Here is a sample of noteworthy items in the article:

Seeking input from customers to help define future directions for its database programs, Hewlett-Packard has released a survey gauging customer needs... "I think that this is a good opportunity to really get into the heart of the TurboIMAGE program and have a voice be heard in a productive, proactive way," said Doug Dedo. "[The survey comes] in a time where it can be incorporated into business planning activities."

The first area [of the survey] involved "Just how do people want TurboIMAGE itself to move into the 21st Century," he said. "We are definitely moving it there... Some people call it mature. Competitors think it is obsolete and yet we've got a phenomenally large customer base and a huge set of applications that are really providing valid business solutions today."

In the second area, HP officials sought input about ALLBASE/Turbo CONNECT write [access], the bridge between the relational and the TurboIMAGE world. "Our ALLBASE SQL product has an ALLBASE/Turbo CONNECT that links it to TurboIMAGE so that you can do an SQL query and be able to pull information simultaneously out of the relational database as well as out of a TurboIMAGE database," Dedo explained.

Within the last six months, customers have expressed a desire to be able to write back into the TurboIMAGE database. "So the second piece in the questionnaire was to start getting more detailed data on what [customers] see and how they would use the write capability."

"The goal of the survey is to compile information on user needs so that these needs can be met by HP in future product releases," Dedo said.

Breaking free from IMAGE's physical limitations

IMAGE's physical limitations have to do with the way HP has chosen to implement (or *not* to implement) its various design criteria. The April 1991 *Interact* includes an article by Wirt Atmar (of QueryCalc fame) called "The future of IMAGE on the HP3000 is SQL." Atmar quotes a senior Hewlett-Packard executive:

"The problem is that TurboIMAGE has been tuned for over 15 years and there are not many ways we can improve it anymore." Wim Roelandts, HP vice-president and general manager of the Computer Systems Group, made this statement last year.

But this is not true. A number of rather simple enhancements to IMAGE would make dramatic differences in its use, in its performance, and in the minds of its users. Now that HP has graciously agreed to implement the critical item update enhancement, the foundation has been laid for a number of truly significant enhancements to IMAGE...

These few enhancements, which would not only revolutionize the use of IMAGE but also ensure its future competitiveness, are among the most commonly touted advantages of SQL databases. But they have nothing to do with SQL per se. They should be part and parcel of any competitive database structure. IMAGE is particularly amenable to these modifications. And none of them are difficult to accomplish. HP already has all of the code in hand to implement each enhancement.

Atmar's article points to the problem *and* to the solution, the implementation of the original ALLBASE idea. Dedo's survey is a step in the right direction. I applaud HP's willingness to give IMAGE the ability to inter-communicate.

Giving IMAGE the ability to inter-communicate

The bottom line is: Because IMAGE needs to inter-communicate with other database management systems, the issue boils down to providing a read/write SQL interface for IMAGE today (whether we like SQL as a lingua franca or not). This will be a significant step in fulfilling the original Hewlett-Packard promise for ALLBASE: Standardized access to IMAGE databases *and* to relational databases. This will provide Hewlett-Packard users and applications developers with the best of both worlds. What a wonderful idea.

(For their thoughtful reviewing, I wish to express my gratitude to Wirt Atmar, Leslie Keffer de Rego, Fred White and René Woc.)

DESIGNED FOR RECOVERY
Paper #3807

JAMES A. DEPP
UP TIME DISASTER RECOVERY, INC.
643 W. STADIUM LANE
SACRAMENTO, CALIFORNIA 95834
916-648-1282

The sage banking president was retiring, and the young teller knew he must act quickly if he were to learn from the old gentleman. Introducing himself, and his desire for wise counsel, the younger man asked, "Sir, how does one achieve such success as yours?"

"Good decisions," came back the terse and sharp reply.

Acknowledging the simplicity of the answer, the young man probed, "But how might one know how to make good decisions?"

"Experience," was the brief reply.

Certain that this made sense, the young questioner pursued his mentor. "But how does one gain experience that he might make good decisions, achieving success?"

"Bad decisions," was the short but telling answer.

We have many opportunities to seek out the advice and experience of others. This is the promise of a symposium such as ICMS. This presentation attempts to use the experience, good and bad, of many people and organizations so that you may benefit without the cost of bad decisions.

We will review quickly the need for having disaster recovery strategies and backup, and the options available. We will review testing strategies and experiences, and extend these to discuss the good practices that grow out of them. Finally we will review some actual disasters and recoveries. They teach us what was not learned in rehearsals.

Need

Does your computer system fail unnoticed? Do users patiently wait until the system is available? Is there no cost to the company for time down?

It is more likely the telephone rings whenever the backup runs a few minutes late, or the software developers compile a program or extract a large block of data. If this is your situation, it is worth considering recovery options.

Studies have concluded that 85% of the organizations were heavily dependent on data processing. Of those responding to surveys, 62% had a recovery plan in place. Despite the plans, preventive measures and security, 31% still suffered service outages.

The cost impact on the company takes many forms. In times of tight money, low margins, or high debt load, protection of the cash flow justifies the expenditure for backup services. Market share is critical for commodity producers who know the customer will simply call another supplier if they are unable to take or process an order. Productivity is always an issue, particularly as staffs shrink. Quality and lot control, whether regulated, or for warranty, or product integrity, are continuous requirements. Legal requirements may demand contingency plans and backup services.

For those who want quantified recovery cost projections, recent studies have suggested considering recovery costs daily as a percentage of daily revenues. The figures can be compared directly to profit margins, and the increase with days of outage can be reflected. The reality is that disasters do occur more frequently than we recognize, and that experience shows that recovery planning and backup system availability is worthwhile.

Recovery Strategies

While it is easy to jump into discussion of hardware options, the people are still the ones who get the work done in most organizations. Recovery strategies should be built around the needs of the people who use and benefit from the computer system availability. These people use applications that vary in importance for different organizations. In general these are:

Accounting and Payroll
Order Entry
Manufacturing
Quality Control
Shipping; Distribution
Planning
Supervisory

By evaluating the types of disasters that may occur and the impact on the staff functions, strategies can be developed for relocating these people, and providing support services like voice communications and information systems. With business functions spread geographically it is also necessary to consider people at remote locations unaffected by local disasters whose needs continue unabated.

Recovery planning limits consideration of people issues to the required applications supporting business functions. While it is likely that not all applications routinely run will be required, some software design may require that less critical data be available for critical functions to operate. It is likely that reduced terminal access will be possible and that printing can be consolidated.

Recovery System Considerations

By defining the staff requirements and locations and the specific application needs, the selection of a hardware configuration becomes straightforward. Access to hardware can take several forms.

Reciprocals and Service Bureaus - Once inexpensive and available, these strategies were not a good solution because testing and control were difficult. Today they are largely unworkable. Service bureaus are increasingly scarce. System size and disc capacity make sharing with another user a double disaster.

Redundant Hotsite - Duplicate, remotely located, available systems provide the greatest control, but at very high cost.

Manual Operations - Systems have grown, and numbers of users increased such that a return to manual operations is rarely possible.

Empty Shell - A prepared but empty location for a computer system is becoming less and less reasonable as an expense for users of minicomputers. Computer processing is needed, and the real need is immediate availability. When a processor is only available at the time of disaster, it prevents testing of recovery strategies.

Vendor Facilities - Years ago it may have been possible for a hardware vendor to make available the demonstration system. Increasingly, vendors have recognized this is unworkable, setting up recovery service options instead.

Recovery Services - By providing the hardware, the environment, and supportive staff, recovery services permit rehearsals of recovery plans. Where bad decisions may be made in planning, the cost of experience through testing remains low. Rehearsals affirm good decisions. Recovery services provide the benefits of redundant systems, while sharing associated costs.

Rehearsal Strategy

All aspects of recovery planning benefit from a stepwise or staged development. This is equally true of rehearsals. Rehearsing a recovery should be a successful venture. It should confirm strengths. This is not a pass/fail test, but an examination of a procedure in terms as life-like as possible.

The most effective rehearsal strategies have defined objectives, and step toward achievement. For example:

1. Load the system and applications, confirming that user access and security is in place, programs will run, and data and other processing resources are available.

2. Establish data communications links so that users can access their applications to affirm their ability to operate in this mode following a disaster.

3. Operate a full business cycle on the recovery system in a scheduled rehearsal at the recovery site or bring the recovery system to the local facility. Affirm that supporting documents and services are available.

4. Simulate the disaster on an unscheduled basis to affirm that all procedures and capabilities operate as required.

Depending on the size of the system, the data storage, and the number of remote users, one or more tests may be required. Often these are four separate events.

Experience

Experiences fall into categories, and a review of recovery procedures against this list presents possible areas for improvement. Specific examples will be given in the presentation.

1) Recovery System Specifications
   - too little disc
   - too much processor
   - overlooked components - PSI, INP, LAN, OPENVIEW

2) System Startup
   a) System Configuration
      - device class names
      - permanent versus transient disc space
      - speed sensing on modem ports
   b) Application Restoration
      - tape directory or files corrupted
      - licensing for backup machine use
      - required files residing outside the account
      - open files not stored during backup
      - lack of prior agreement on which applications
      - required equipment not accessible
   c) Security
      - passwords for security packages not recorded
      - port specification in security packages

3) Data Communications
   - system and multiplexer port identification
   - complete cable specification

4) Data Update
   - available source documents
   - time to update versus use of logging
   - all required hardcopy information on computer records

5) Staffing & Communications
   - too many 'hats' to wear
   - insufficient user staff at recovery site to handle rerouted incoming telephone calls
   - insufficient advance communication throughout the organization on the critical nature of recovery.

6) Procedures
   - dump procedures fully inclusive of files
   - tape testing and cleaning to ensure quality media

7) Return To Live System
   - protection against tape errors
   - retention of full dump from destroyed system
   - procedure for merging files from backup system
   - application of log files

Experience from Recoveries

Two recent disasters illustrate the need for planning and rehearsal, and point out pitfalls if rehearsals are not complete. The bank fire is notable in that it displaced numbers of people. The head crash points out the need for tight procedures.

The bank had planned and rehearsed, with rehearsal less than two months prior to the disaster. The fire was not the most likely disaster. The aftermath included 4 different moves, each slightly more devastating to morale.

The early days of disaster are full of camaraderie, sharing of office space, and tolerance for each other. As time wears on, so due nerves. This company did an excellent job of supporting staff with small groups, counselors, and encouragement.

The head crash was serious, but when the only backup tapes came back from offsite storage all the managers were meeting to review strategy. The staff changed shifts, and HP had already repaired the problem. The routine was to use the tapes from the rotating offsite canisters for the next dump. The canisters were then returned to storage.

Since the operator was not told anything different, and because he did not know of the problem, it was reasonable to take the tapes and do a dump - of a crippled live system on top of the only current backup of that system.

Bad decisions, good decisions, opportunities to protect the business, and maintain profitability. Experience.

PAPER #3808
GETTING THE MOST OUT OF IMAGE

C. BRADLEY TASHENBERG
PRESIDENT

BRADMARK TECHNOLOGIES, INC.
4265 SAN FELIPE, SUITE 800
HOUSTON, TEXAS 77027
(713) 621-2808
FAX (713) 621-1639

When Hewlett-Packard decided in 1972 to provide a network database on the HP 3000, it did its homework well. Back then and well into the 1980's the most widely used and sought-after database tools were the network database products. What makes HP unique is that in making its commitment to offer IMAGE, it looks a position of prominence in being the first (and for years, the only) vendor to offer a vendor-supported database manager in the minicomputer market.

The success if IMAGE in the past 15 years has made users wonder what the real limits of IMAGE actually are. In looking back, I first got involved with the HP 3000 at the time of the CX. At that time the HP 3000 was a very limited machine, and, correspondingly, the databases on it reflected the system's limitations. However, over the years, as the HP 3000 systems have grown more powerful, the database sizes and user environments have grown in geometrical proportion. Today it's not individual sets within the base may hold several million entries. IMAGE has supported ever growing databases practically since its inception. This speaks well for HP, but one must eventually ask, "Where do the real limits lie?"

Well, about six years ago I received a call from one of my customers who informed me that, in attempting to do a capacity change on one of his larger sets, the capacity change feature did not work successfully. I, of course, was very concerned, but before I panicked I asked him what error message appeared at the time. He told me the routine stopped just after attempting to build a new dataset. The message that was displayed from the utility stated "attempt to build too many sectors in an extent."

From this message I could tell the routine had attempted to create a file with the maximum number of extents and the maximum number of sectors within each of the extents, and had failed. This made me wonder what had really happened. I, therefore, asked him how large the original set was, how much was he trying to expand it, and what the block size of the dataset was. He said the set had 2,000,000 entries, that it had a block size of 512 words, and that he was trying to expand it to 2,250,000 entries. I told him to give me 5 to 10 minutes and I'd get back to him after I figured out what happened.

After hanging up the phone, I took a notepad and started calculating. I knew that all IMAGE datasets had to live within the limitations of the MPE file management systems. I also knew that no MPE file could be greater than 32 extents, and that no extent could have more than 65K sectors. So, to make any meaning out of this, I decided to reduce these limits into an IMAGE environment where the physical block size of the dataset was 512 words (see Figure 7).

Then I phoned him and asked one simple question. I asked if his blocking factor on the set in question was "4." In astonishment he replied, "Yes. How did you know?" The tragedy behind this scenario is that he was in a manufacturing shop with more than 100 users on the base, and that he couldn't allow them back on until he could expand the set.

In discussing the problem with him further, I ascertained that because of the size of the media record, if we were to increase the size of the block to 1024, the blocking factor would change to "9" and he would be able to make the capacity change. Since he had no alternative, he did just that and it worked.

Unfortunately, by increasing the block size, I told him that he would probably lose 10 to 15 percent in performance, since the internal buffers of IMAGE are based on the largest block size. Till then, his largest block size had been 512 words. Now it was 1,024. Therefore, by doubling the block size to make the capacity change, he cut his IMAGE buffers in half.

I guess one would call this a double whammy! To me, the unfortunate thing is that he had no idea he had already reached the file limit. In the IMAGE Manual it states no set can be greater than 8 million entries, and, in the TURBO Manual it eliminated this restriction altogether. So, what was he to think?

What I have found over my years in the computer business is that it is not the difficult things that get you, but the simple ones. And, of course, the corollary that goes hand in hand with this that it's not what you know that gets you, but what you didn't know.

These statements apply fully to IMAGE. When I first was involved with IMAGE and went to the IMAGE class, the attitude was that I only needed to learn how to write an IMAGE scheme, how to use the IMAGE intrinsics, and how to use Query. The rest was IMAGE's responsibility.

This is all well and good. But it isn't true. IMAGE isn't a black box like a modem or a video terminal. You can't just throw it away or replace it if it doesn't work to your satisfaction. You've got to live with it. It can be an asset to you or a liability. And, the difference is all in the way you use it. It stands to reason that the more you know about IMAGE the better you'll use it.

Inconsistencies And Idiosyncracies Within IMAGE

What I found most interesting about IMAGE is its little inconsistencies. This is not to say that IMAGE is not a good product. Quite the contrary, it is an excellent product. But all products have their idiosyncrasies and IMAGE is no exception.

Take DBSCHEMA for example. One would think that it would be totally compatible with the rest of IMAGE, when, in actually, it's not. DBSCHEMA, DBUTIL, CREATE wants to optimize disk space. So even though it uses 512 words as a default block size, it may very well assign block sizes of 384 or 640 words, or whatever size it deems necessary based on how efficiently the space can be allocated. Therefore,after DBSCEMA, DBUTIL finishes its processing, it's not unusual for a database to demonstrate total inequality in the sizes of its various datasets (see Figure 2)

Now, looking at it from a runtime perspective, IMAGE requires internal buffers must be large enough to accommodate the largest dataset block size within the database and specs its buffers to that size.

For those sets with block sizes conforming to that size, there is no problem. So when IMAGE performs an I/O to those datasets, the buffers are fully loaded. However, for sets that have smaller block sizes, the buffers are only partially used. This not only wastes space but also requires additional I/Os.

Take for instance the previous OE database example. Notice that the largest block size in this base is 640 words. This means that the internal IMAGE buffers will be spaced at sizes of 256, 384, and 512 words?

For the set with a block size of 256, less than half of the buffer is being used each time an I/O is performed. So if the block size were expanded even to 512 words (which is not the most efficient) on a serial read of this set, it would require half as many I/Os to read the set. If this were a detail set, then on a chained read it might minimize the online I/O processing by as much as 10 to 50 percent.

For those sets sized at 384 words, 40 percent of the buffer spaces are being wasted and a comparable I/O burden is being placed on serial reads of these sets. Again, if these sets were detail sets, equivalent degradation could take place on chained reads of lengthy chains.

The moral of the story is that irregular block sizing is not good! This is a common problem with IMAGE because of the inconsistency between DBSCHEMA encourages irregularity in block sizing because it wants to optimize disk space, while the intrinsics want regularity in the buffering. Therefore, do a LISTF base @, 2 on your database and see if you suffer from irregularity. If you do, reblock the sets to a common size and work with the system rather then against it.

A final point to make on this topic is that if you find a set that has a particularly large block size within your base, know the effect that set is going to have on your performance. If at all possible, try to cut back the size. If you can't cut back the size, see if you can place it in its own database. Though that may seem like a radical solution, let's examine the options. The downside is that by placing the set in its own base, an additional DPOPEN will be required whenever you need to access it; and, of course, the base will have to be added to the backup process. The upside is that you will not only gain more buffers by removing the set from its original base but you will gain an additional set of buffers to use in processing it. So in retrospect, the solution is not such a radical one after all.

BUFFERING

Buffering brings up another interesting point. In IMAGE, there is a trade off between the size of the IMAGE buffer and the number of buffers that can be internally supported. The larger the buffer size, the less the number of buffers IMAGE has at its disposal. The smaller the buffer size, the more internal buffer that can be assigned. The trade-off is that a larger buffer size can accommodate a larger block and, therefore, minimize the I/Os required to process through the dataset. The less the number of buffers, the greater the possibility that IMAGE will run out of internal buffers during online processing, and, in so doing, need to swap out active buffers to make room for current I/O requests.

Therefore, the ideal situation is to find a standard buffer size that will accommodate a reasonable amount of buffers. Figure 3 shows how dramatically the number of possible buffers accommodate larger dataset blocks.

As stated earlier, DBSCHEMA uses 512 words as a default value for the standard block size. In studying many cases, I have found this to be an excellent value. It offers a sufficient size to house a reasonable number of entries and yet it is small enough to yield a sufficient number of buffers.

For TURBO users the DBB control block, which under the classic systems was constrained to an extra data segment, has a 32K upper limit in which to house its IMAGE buffers. So if your system is a classic, I think you will find better performance out of IMAGE if you keep your standard dataset block sizes around 512 words.

On the other hand if your system is a SPECTRUM, the DBB control block in which the buffers reside has been placed in a mapped file where the 32K limit is artificially imposed. However, being a mapped file, this DBB has no real 32K limitations and, restriction shall be lifted. Therefore, for those of you using TURBO/XL, it is extremely probable that you will be able to expand you IMAGE block sizes with out affecting the amount of internal buffers allocated to the environment.

IMAGE Defaults - Not Always Good

Generally, defaults are very helpful since they allow the user to avoid making a decision that could be a bad one. But, in the case of assigning internal buffers for IMAGE, the default is anything but optimal. If you ever ran DBSCHEMA, DBUTIL, CREATE and checked the buffering specifications that IMAGE assigned, you would find something like this:

8(1/2), 9(3/4), 10(5/6), 11(7/8), 12(9/10), 13(11/12), 14(13/14), 15(15/16), 16(17,/18), 17(19/120)

This is telling you that for one or two users, IMAGE will assign eight buffers; and for three to four users, IMAGE will assign nine buffers; and for five to six users, IMAGE will assign ten buffers. This continues until you hit 19 users. At 19 to 120 users, IMAGE will assign 17 buffers.

This means: (1) no more than 17 buffers will ever be assigned for this database and (2) every time a DBCLOSE is executed IMAGE will reshuffle its internal buffers to correspond with the buffer specifications. Both of these practices are extremely inefficient and severely limiting. In databases where the largest block size is around 512 words, there is sufficient space in the buffer area to hold as many as 50-60 buffers. So, why would you want to constrain the buffering to only 17 buffers. And likewise, where you have up to 75 users signing on and off the database, why would you want to go through the unnecessary overhead of shuffling the buffers around in the control block? Both practices are inefficient and uncalled for.

To avoid this pitfall, all you have to do is use the SET BUFFSPECS command in DBUTIL and reassign your buffer specifications to 200(1/120). This, in essence, tells IMAGE you want 200 buffers to be assigned whether there is one user or 120n users on the database. Obviously, 200 buffers won't be assigned but IMAGE will allocate as many buffers as it has room for in the DBB. The other benefit to this specification is that IMAGE will allocate as many buffers as it can get and never reshuffle, no matter how many users are using the database. This simple one-time specification could have the effect of improving performance by as much as ten to 15 percent.

WHAT Else You Should Know About IMAGE

Much of what I've learned about IMAGE I learned through experience - at my own and that of my customers. And, as I said before, it's not what I knew that got me in trouble, but what I did not know. Or more accurately, what I did not expect. Running across a dataset upper limit that you don't expect is extremely annoying. It raises the question, "what other things do you know?" So many things are left unanswered, such as: Blaming poor performance on the system when the problem might be something as simple as poor buffer assignments. Or wondering why IMAGE performs poorly on some sets and well on others. These can all be mysteries if you don't understand the internal workings of IMAGE.

Those kinds of incidents made me wonder what more I should know about IMAGE, and caused me to launch a further study into it over the past ten years. I realized the more I understood about IMAGE, the easier it was for me to cope with what I was experiencing . So, I broke my study into two segments. The first segment dealt with MASTER sets. How they were built and how they behaved. The second dealt with DETAILs.

Myths About MASTERS

Over the past 15 years I had been told many different things about MASTERS. Some proved to be; some didn't. One thing I did learn from the experience was not to believe everything I was told. Many things I was told were not misleading, but just plain wrong!

Myth 1 - Never Use Numeric Keys

The most misleading advice I have heard over the years is not to use a numeric key, because if has a tendency to generate too many secondaries. Well, this may be the case in certain circumstances. But it isn't the case in all circumstances. You need to know only how the algorithm for numeric keys works to realize there are some excellent scenarios in which you wouldn't want to use anything other than a numeric key. Numeric keys use a modular function as an algorithm. In doing so the key value is divided by the capacity of the set, and the remainder resulting from the divide is used as the entry location. So, if the key value is always less than the capacity value, the algorithm will always generate a unique entry address, and therefore, you'll never see a secondary in this set in your life. However, in sets where the key value us substantially larger than the capacity value, there is a very good possibility that a large number of secondaries may result because the algorithm is continuously generating the same remainders over and over again.

So if the myth is to be accurately stated, it should be modified to say, "if you capacity is considerably smaller in size than your key value, don't use a numeric key. "Otherwise, if your key values are less in size than your capacity, don't hesitate to use a numeric to use a numeric key.

Myth 2 - Always Allot A Sufficient Amount Of Free Space In MASTERS

In going on-site to various locations I commonly find that many IMAGE users allocate excessive amounts of space in MASTERS. When I ask why, the usual answer is that they are trying to minimize the amount of secondaries within the set by increasing the free entry size. Though this is a noble effort, it usually causes more problems than it solves.

Take the example of a hypothetical database that has a MASTER with a blocking factor of five and a capacity of 4997. Let's also assume the key has been defined as alphanumeric. As shown in Figure 4, if we were to place three entries into this set (assuming the set was completely empty when started), it is conceivable that the first entry could end up two-thirds of the way into the set, the second entry in the first block of the set, and the third entry in the last block of the set. The reason for this is that alphanumeric keys use a hash algorithm, and that the function of the hash is to distribute the entries as evenly across the set as possible. So, the algorithm was just doing its job.

But what effect does this have on performance when we want to perform a serial read of this set. The answer, of course, is that it has a substantial effect. For, in order to read those three entries IMAGE must perform 1,000 physical I/Os. In pre-TurboIMAGE, even if all three entries happened to hash into the first block of the set, IMAGE still would have performed 1,000 physical I/Os. Because IMAGE reads to the high-water mark in serial reads, and the high-water mark for MASTER is the capacity. This has been changed under Turbo, but even with a few dozen entries there is still a high probability that one of these could find its way to the bottom of the set.

So the lesson to be learned here is not to allocate excessive amounts of free space in MASTER datasets, because it imposes a substantial processing burden on IMAGE during serial reading. For those of us who use 4th-generation languages, realize that most of these languages perform a tremendous amount of serial reading.

Of course, the main reason people allocate excess space is to limit the amount of secondaries within their MASTERS. We all know that as a MASTER dataset begins to fill up, there is an increased probability that the hash algorithm will begin to hash to locations already in use. When this happens IMAGE has to find a new home for the secondary. If it can find a location within the same block as the primary, even though a secondary chain is formed, the creation and lookup of the secondary is not inefficient from an I/O standpoint. It takes one I/O to process the primary, and needs no more to process the secondary.

However, if it can't find a location within the same block, IMAGE becomes performance inefficient. It must read adjoining blocks until it finds the first available location. If the hundreds of I/Os until it finds a location. This is why users want to minimize the amount of secondaries within their MASTERS.

What is even worse is that after the set begins to gather secondaries, there is a possibility new entries being added to the set may hash a location already being used by am secondary. In this event, since these secondaries have been arbitrarily located, the hash takes precedence and the secondary must be relocated so that the new primary can be entered. These are called migrating secondaries.

Examining Figure 5, you can see the effect secondaries have on the loading of entries into a MASTER dataset.

At the beginning of operation the entries go into set quite quickly. But after a period of time the secondaries start to slow up the process. As the number of secondaries grows, secondary migration takes its toll and the operation really starts to degrade.

So avoiding or minimizing secondaries within the set is the best answer in preserving the performance of a MASTER dataset. And the solution that had been proposed many years ago to solve this problem was to use a primary number as a capacity.

Myth 3 - Always Use A Prime Number For Your Capacity

In all the studies we have done the last five years, there is no evidence to substantiate this myth. As a matter of fact from all the testing I've done, I have found that the hash algorithm favors non-prime numbers rather than prime numbers. But this is not such a surprise. One of the HP lab people wrote an article in the Anaheim International proceedings back in 1984 stating the same thing. All I did was perform empirical tests to determine if there was any truth to the myth.

My results were quite interesting. Though I found no validity in the myth when it came to alphanumeric keys (which use the modular function). Here again, the myth isn't totally wrong. It's just quoted out of context.

However, of all the techniques I have explored to find an optimal capacity value, the one that seems to work best is the capacity sampling technique. The concept behind this technique is to extract a large sampling of keys from the MASTER. Then, based on a range value, sample the capacities within the immediate range (+10) to determine which of these capacities generates the least amount of secondaries and offers the best distribution. Because of the sensitivity of the capacity values within a narrow range, generally one or two stands out well above the rest. And that is the one you want to use for your new capacity. Two benefits result from this approach. The first is that the capacity change runs faster. But the secondly and more important is that the performance of the set is optimal after the set is turned back to the users.

DETAILS

In MASTERs, the problems seem to orient themselves around wasted free space, excessive secondaries, migrating secondaries, and poor entry distribution that result in clustering and degrading performance. DETAILS have different kinds of problems, relating to the effects of chaining within the set. Let's take an order entry system for example. Assume in this example there is only one order entry clerk responsible for entering orders, that the average size of an order is six line items, and that the blocking factor of the order line DETAIL set is three. Suppose the set had been completely empty. As the clerk entered the first order, the first three order lines would go into the first block of the set and the next three order lines would go in the next block. Therefore, if the clerk decided to review the order directly after the order had been entered, it would take only two I/Os for those six order lines to be extracted by IMAGE.

However, this isn't typical, is it? Generally you have a group of order entry people processing orders. And, in this scenario, assuming they all type at the same speed, each of the order lines will end up in a separate block from the other lines. So when it comes time to review the order, it will take six I/Os to extract the six order lines instead of two I/Os. Considering the original examples as an optimal situation, this means that in this environment just by attrition, IMAGE is 200 percent inefficient in its processing of entries.

Though this may seem bad, in a cache domain of 4K-16K, the inefficiency may be purely academic since the other entries probably reside in adjoining blocks, which are more than likely in the cache domain.

However, let's expand this scenario one step further and assume that just prior to the entry of new orders, a daily or weekly maintenance run was performed that deleted all closed orders and left a lengthy delete chain in its place. Now when the new order lines are entered into a set, since IMAGE requires that the delete chain area be used before entries are added to the end of the set, these new order lines will be spread throughout like a can of worms. Therefore, due to the sporadic nature of the delete chain, there is very little likelihood that these new order lines will be found in the cache domain. So, in this case, inefficiency has no redeeming grace.

In very large DETAIL sets, you normally have a combination of both circumstances occurring, and based on the size of the average detail chain, you may periodically be required to reorganize or repack the set. Usually the first time you do it you experience a dramatic improvement in throughput. Because of the tremendous success of the first one, many people have made the mistake of immediately scheduling another reorganization the following week or month, only to experience no noticeable improvement in throughput. However, had they performed an inefficiency study, they probably would have found that the new transactional activity since the last reorganization was not that substantial, and that the set was only ten to 50 percent inefficient. In I/O terms, if a set is 10 percent inefficient it requires 1.1 I/Os when it should require only I/O; and likewise if the set is 50 percent inefficient it requires 1.5 I/Os when it should require only one I/O.

So, before doing a reorganization make sure that the inefficiency ia at least 100 percent, which yields a 2:1 ratio of actual I/Os to optimal I/Os. If it is less, the set is not a candidate for a reorganization, because after the whole process is over, the user won't even notice the difference.

Primary Path Assignment

You might have wondered when you built your schema why the schema processor required a primary path for DETAIL sets. Well, the reason is quite simple. Since the reorganization process can't recondense the DETAIL set on all paths, it uses the primary path as its selection. Many times this path is poorly assigned and in some instances it is even assigned by default.

The problem with IMAGE assigning the primary path by default is that there is no scrutiny involved. It just goes ahead and assigns the first one it encounters, and in so doing potentially could assigns the worst from a standpoint of efficiency. Therefore it is important that the primary path be properly assigned by the database designer. Defaults aren't always good.

To good maximum benefit from the primary path make sure the path is the most frequently used path, and that the average chain length is of a reasonable size. If the path is not frequently used, then why make it a primary path? Select another. You'll get more benefit out of it. Likewise, if the path has an average chain length of one or two, there is absolutely no benefit in making the path a primary path. For a path with a chain length of one, it takes at least one I/O to retrieve the DETAIL; so, that path will never have an inefficiency. For a path with a chain length of two, the worst situation would result in a maximum inefficiency of 100 percent. So try to place your primary paths on paths that have chain sizes greater than two, where they can do the most good.

Effect Of Blocking Factors

In talking about efficiency, another key factor that can dramatically effect the efficiency of path is the set's blocking factor. The purpose of a blocking factor is to allow multiple IMAGE entries to reside in an IMAGE block. If there were only one entry in a block (which is equivalent to a blocking factor of 1), the entire concept of I/O efficiency would be academic, since each entry would require its own I/O. However, if multiple entries can be read through a single I/O, the larger the chain and the larger the blocking factor the more efficient the performance. This is the environment where the reorganization or repack will do the most good.

I've seen databases where the relationships between the paths, though not immediately apparent, resulted in improved performance by altering the primary path from one path to another until an ideal path was found. This normally will not work in environments where there is mutual exclusivity or independence in the functionality of the various paths. But in many circumstances, paths have dependency relationships, as in our example of the order entry system. In this example it wouldn't be unusual to have a path on the order number, one on the date, and one on the part number. Though none of these seems to be directly related to one another, in reality the order number and date are related by chronology. Order numbers are usually sequentially assigned, and the range is usually governed over a certain date. So placing the primary paths on the order number could actually improve the performance of the date path, though neither would have an effect on the part path.

Therefore, look for natural relationships of this type in the selection of your primary path assignments.

Unused Paths

All of us have designed packages or systems where we incorporated paths within the database that were for future use or for completeness of the package. However, many times these paths (or the features they represent) are never used. The user does not want to change what has been working successfully. And because it's a package, there is kind of an "if it works don't fix it" philosophy. But the real issue is: Are they harmful?

The answer to this question is unequivocally "yes!!!" Without question, paths not currently being used cause tremendous degradation within the online processing environment. It takes three to four I/Os to maintain the pointer changes on each and every path of every entry that goes into a DETAIL set. And if the maintenance requires an update of the MASTER chain head, the MASTER is on a lengthy secondary chain, the process will take even more I/Os.

From a performance standpoint, therefore, unless each DETAIL path is an essential path, it should be removed. I mean essential from an online processing vantage point. So if in reviewing your database you find a path being maintained just to ease the access of information for a weekly or monthly report, remove the path. This will minimize the overhead and IMAGE without substantially adding to the time required to prepare the weekly or monthly report. The net effect is generally an overall savings in processing time.

Recapping DETAILS

So in reviewing DETAIL sets from a performance standpoint, the major issues are:

1. **Proper selection of the primary path.**

   If your primary path is not an effective path, then all the reorganization or repacking in the world won't improve the performance of the set.

2. **Decent blocking factor size**

   If the primary path has been set but the blocking factor is so small that it is not accommodating enough entries, then here again, you won't be able to improve the performance on the set too effectively.

3. **Eliminating sort chains**

   Through we did not discuss sort chains or sort paths, in almost all instances they are bad for performance and should be removed if at all possible.

4. **Avoiding creation of delete chains**

   Delete chains are often a necessary evil within your DETAIL set. From a performance standpoint, they are a disaster. So if you're doing daily maintenance, consider flagging the entry for deletion instead of deleting it. Then at the end of the month delete the flagged entries.

   For better results, perform a DETAIL reorganization directly after the deletions. This will eliminate the delete chain entirely.

5. **Eliminating unused paths**

   If a path is not being used or is used infrequently, remove it from the dataset. It's imposing an unnecessary processing burden.

6. **Reorganizing the set only when necessary**

   Reorganizations will help performance only when the chaining along the primary path is so inefficient that it's causing more than a 2:1 ratio in I/O performance. If after doing an analysis you find that the inefficiency is less than 100 percent, the effect of the reorganization will only be marginal.

Summary

In summary, IMAGE has been one of the most successful database tools in the computer industry to date. For many years it has earned the coveted DATAPRO award for software excellence. It has survived 15 years in an industry that makes products obsolete in less than five year, and it shows no signs of weakening. But like any product, to get the most out of it, you must use it properly.

I have listed many of the common things you should be aware of to take full advantage of IMAGE. Most of them are simple and, for the most part, easy to implement. What I like most about dealing with IMAGE is that in may cases you don't have to redesign your application system or modify a single program. All the things I touched upon can be done directly to the database with HP or third-party utilities.

So take advantage of some of these recommendations and you be the judge. I will be shocked if you don't improve the performance of your base by 10 to 30 percent.

FIGURE 1: IMAGE DATASET

Block Size	Blocking Factor	Actual File Limit (entries)
512	1	500,000
512	2	1M
512	4	2M
512	8	4M

FIGURE 2: IRREGULARITY IN DATASET BLOCK SIZES

Filename	Code	Logical Record Size	Type	Sector X	MX
OE 01	PRIV	128W	FB	1	1
OE 02	PRIV	384W	FB	1	1
OE 03	PRIV	512W	FB	1	3
OE 04	PRIV	384W	FB	3	3
OE 05	PRIV	256W	FB	1	1
OE 06	PRIV	512W	FB	1	1
OE 07	PRIV	384W	FB	1	1
OE 08	PRIV	512W	FB	32	32
OE 09	PRIV	512W	FB	3	3
OE 10	PRIV	512W	FB	1	1
OE 11	PRIV	384W	FB	32	32
OE 12	PRIV	640W	FB	32	32
OE 13	PRIV	512W	FB	5	5

FIGURE 3: INTERNAL BUFFERS

(ASSUMING 30,720 WORD OF SPACE IN THE DB)

Block size	# of Buffers
512	60
1024	30
2048	15

Paper # 3809

Effective Vendor Relationships

by

Alan Young
% Bekaert Corp.
1395 So. Marietta Pkwy.
Marietta, Ga. 30067

INTRODUCTION

Effective relationships start with a solid foundation of marketing. Marketing skills are not given or passed on through our genes, they are acquired through learning and understanding. Experts in the field of marketing spend years to understand four basic things:

1. The market
2. The customer
3. The competition
4. The product

Successful marketers divide their time and skills equally between these four basic concepts. This paper will focus on the most elusive concept, i.e., the customer. By presenting both sides of the relationship we can explore ways to better reach the customer. Starting with the vendor, we will examine the selling through the eyes of the company and the sales rep. Then we will discuss the customer through their business and personal needs.

THE VENDOR

"Rarely does a single thing influence a purchase decision. More often, it is a series of items that cause prospects to form opinions."

I will not attempt to identify all the various techniques used in marketing. Most of them can be found in any marketing textbook in your local library. But taking a look at some of the more obvious ones is necessary to avoid over-looking the importance of these techniques.

IMAGE - Does your company's letterhead, colors, catalogs and brochures demonstrate an air of importance and success? Are you using glossy prints or just plain photocopies of material? Do you hand out price list and color catalogs or do you scribble information on a note pad when you finally get to meet with your prospect?

NEATNESS - Neatness closely aligns with image. Don't display sloppiness on your premises or in your business practices. There is often an unconscious but natural assumption that if a company is sloppy in any way that must show up in how they run their business. Look around and test yourself. Is your office neat and well organized or is it messy with lots of papers on the desk? Does your office present an image of someone who is confident in their ability and skills or someone who is disorganized and easily ruffled? Are your premises clean and friendly or dark, dirty and foreboding? Would you invite someone into your home if it had the same appearance as your business premises? What about your delivery vehicles and company cars? That sales rep, who is attempting to close deals involving hundreds of thousands of dollars, is his car beat up and old or does it have the kid's cookie crumbs on the seat along with hairs from the family dog? Finally, take another look around the office or premises and pay particular attention to the signs. What image do the signs convey? Do the employees have jokes and messages hanging about? Is the art work tasteful and classy or are the pictures cheap prints and posters?

DRESS - Dress standards have been relaxed over the years and some people will have you believe appearance is not important. Do not believe them for one moment. Sales reps and other employees who deal with the public are constantly presenting your company's image. When you fly on a commercial airline does the dress code of the flight attendant present a first class business image or would you prefer to see them be able to wear any old clothes, wild hairstyles and excessive makeup. Does your salesman similarly project an image of success or is their dress indicative of a maverick conveying an attitude of indifference? Dress and images alone does not guarantee success but do I need to elaborate on the importance of first impressions?

Once you are convinced you have the right calling cards and you begin to work on converting prospects to customers, ask yourself this question. "Would I want to do business with my company if I were on the other side of the desk?" Assuming you have a good product that customers desire, can they do business with your company? Do your credit and administrative procedures make your company difficult to do business with? Do you have a good record of on time delivery, available merchandise and consistent quality? Have you even targeted the proper market and do you understand this market?

THE CUSTOMER

The surest way to improve vendor relationships is to stop treating your vendors as adversaries. A cooperative effort will help each other succeed in the marketplace. As the customer, you have a need. If the vendor can fulfill that need then that vendor has climbed over that first hurdle leading toward a happy relationship. Figure 1 shows some of the hurdles that, once overcome, constitutes a healthy relationship.

Figure 1-1

Vendor vs. Company Agendas

Vendor	**Company**
Desired Product	Desires product
Product Quality	Satisfaction level (Inherent)
Responsiveness/Feedback	Satisfaction level (Achieved)
Competitive/Informative	Price Effective
State-of-the-Art	Progressive
Expanding Scope	Growth
Salesmanship/Problem Resolution	Personal Attention

Company:
The company **desires** a product. The product could be materials, supplies, capital equipment or a service.

Vendor:
The vendor wants to provide a **desired** product. Vendors do not advertise their products as being "almost-as-good-as" or "4th in the market place". Every vendor tries to offer you a benefit of doing business with them.

Company:
The company prefers to buy goods that have an **inherent satisfaction** level. Hewlett Packard has been manufacturing equipment for a long time that has proven reliable and works for a long time between servicing.

Vendor:
The vendors are emphasizing **quality** in their manufacturing operations. They develop slogans and fully support quality control programs.

Company:
The company continues to buy goods that have an **achieved satisfaction** level.

Vendor:
When the vendor is responsive in assisting customers with problems and provides the customer with feedback, the customer satisfaction level is raised beyond the basic quality of the product. "Feedback" is what separates the vendors who show a caring attitude from vendors who simply want the fast buck. Feedback may be as simple as making the customer aware of price increases, model changes or policy changes, or feedback may be involving yourself in a search for an item the customer needs that you don't sell.

Not all vendor relationships work on a controlling premise, nor do they always have to. Some activity merely involves a basic transaction or a "one time buy". In these cases, we are not necessarily looking for that personal touch or lasting relationship. A good product acquired at a fair price may be sufficient in consummating a deal and enough to bring us back in the future. A major concern in dealing with any vendor is getting the best price and a fair price. We our certainly concerned about prices in our personal lives and hopefully we carry that attitude with us into our business dealings. If we read the newspaper and discover an advertisement for an item priced lower than we anticipated, we want that same price even if we deal with a vendor other than the one who advertised. Hopefully we are realistic in what we expect to pay for a product or service realizing that the vendor must make a profit to remain in business.

VENDOR - CUSTOMER RELATIONS

According to the federal Office of Consumer Affairs, 96 percent of all dissatisfied customers never bother to complain to the seller. They make their dissatisfaction known in other ways. Ninety percent of them will take their business elsewhere. Consumer complaints should not be seen as a threat to your business. Look at them as an opportunity to strengthen your marketing efforts.

"Marketers at Ford Motor Company have discovered that if they could keep loyal customers in a lifetime relationship, Ford would realize a $140,000 profit per customer", says Ray Considine, a business consultant in Pasadena, California. As a seller, you should strive for creating a lifetime contract with your customers. Although Ford is a large company with a healthy marketing budget, good customer relations affect the revenues of all businesses. The American Management Association estimates that 65 percent of the average company's future business comes from its present customers. It makes good dollars and sense to keep your current customers. According to the Technical Assistance Project of Washington, "the marketing costs of attracting new business are usually five times greater than selling to existing customers".

Although IBM has a huge marketing budget, one simple goal - good customer relations - has played a key role in making it an international business giant. F.G. "Buck" Rodgers, former vice-president of marketing for IBM and the author of "The IBM Way", says, "No magic formula or guarded secret keeps customers 'married' to IBM long after their equipment is installed and their check deposited. IBM simply approaches the customer, after the sale, with the same interest and attention as when he was the prospect being courted."

CONCLUSION

As a customer, I like to see in my vendor the same qualities that I liked to see in the Sales Reps who I worked with when I was a vendor myself. The successful Reps shared five things in common:

1. They loved to sell.
2. They enjoyed people.
3. They had a sincere and high degree of enthusiasm.
4. They knew their product and their prospects.
5. They believed they would close every sale.

I believe these five qualities are common in first rate sales reps working for first rate companies who are the first rate vendors that I choose to call "my vendor".

BIBLIOGRAPHY

J. Donald Weinrauch & Nancy Baker.
The Frugal Marketer. New York: AMACOM, 1989.

Levinson, Jay C. Guerrilla Marketing Weapons.
New York. Plume Publishing, 1990.

Mroz, Ralph. The Formula for Successful Marketing.
California: Avant Books, 1990.

PAPER NO.: 3810

TITLE: Quick Access to Archived Data for the HP 3000

AUTHOR: Husni Sayed
IEM, Inc.
P.O. Box 8915
Fort Collins, CO 80525
303-223-6071

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO.: 3903

TITLE: The Next Step in Data Center Automation: Switching From Pre-Printed to Laser-Printed Forms

AUTHOR: Judy Frels
Tymlabs Corporation
811 Barton Springs Road
Austin, TX 78704
512-478-0611

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PC Electronic Software Distribution and Management

By Garry Orsolini

Hewlett Packard
Software Technology Center
Roseville, California
916-785-4624
ICMS: March, 1992

Trying to Manage the PC Software MADNESS ?!?!?

It took 40 people working 12 straight hours to upgrade WordPerfect on 800 stand-alone PCs at the New York law firm of Weil, Cotshal & Manges. [1]

At Travelers Insurance Company in Hartford, Connecticut, Don Rindfleisch still vividly recalls the nightmare weekend he spent upgrading an application running on 400 workstations. "We had to send a team of technicians to four different locations and have them swap out software during the weekend. The process was inconvenient and expensive". [2] In another department at Travelers, it took 5 engineers 2 months to perform a single software upgrade on 350 PCs. [3]

And at Siemens Medical Systems in Iselin, New Jersey, 2500 employees using 1000 desktop PCs and 400 laptops, continue to grapple with the question of undertaking company-wide software updates. According to Joanne Witt, a Siemens senior technical analyst, "at what price are we not making a particular upgrade?" [4]

These examples represent the rapidly expanding problems faced by corporations trying to distribute, install, track, and manage PC application software. These problems focus on centralized PC software administration. Additionally, when companies lack a central point for PC software administration, the problems are magnified because end-users are held responsible for PC application installation and upgrades.

Whichever the case, a company's bottom line is adversely impacted. This paper explores these issues in detail, as well as the underlying solution framework for centralizing and automating the distribution, installation, and management of PC application software.

Electronic Software Distribution - A Better Mouse trap

For a long time, companies have been clamoring for a solution to this dilemma. The answer may have come with the advent of Electronic Software Distribution. ESD, in its simplest form, provides the ability to transfer application software and/or data to a mainframe or server, and then distribute the applications and data to PCs or file servers over LANs or other connections. PC users generally interact with the process and their involvement ranges from answering simple questions or entering passwords to supplying additional software disks. This type of ESD installation is called *network-enabled*.

A second method called *tool-assisted* ESD, a superset of the first, has additional capabilities. The tool-assisted method provides a scripting language that facilitates the complete packaging of an application on the mainframe or server. Because the scripting language allows the application to be custom-packaged for a designated PC environment, user interaction is virtually eliminated. Software installations or upgrades can be either pulled or pushed. First, the user can "pull" the software off the server by a simple "press of a button" (very similar to a soda vending machine). The second option gives a PC software administrator the ability to push the software from the server to every networked PC, ensuring proper installation given a PC's specific configuration.

The third ESD approach involves *network-aware* applications. This technique places one copy of the application on the network and allows a network connected PC user to run the single copy. Typically, any data generated is saved on the user's own PC disk. Installation and updates on individual PCs is eliminated. However, files unique to each PC are typically downloaded to the PC using one of the two methods described above. Although there are currently few applications designed to run network-aware, they are growing in popularity. There are still many unresolved issues surrounding network-aware applications, such as licensing, concurrent usage, performance, availability, and security.

ESD Has Come Of Age

According to Eric Hindin, Data Communications Publications, "ESD has been one of those technologies that vendors and users have been all too willing to relegate to the back burner... but finally [ESD] is ready to take its place as a mainstream technology". [5] Apparently, there is strong demand for ESD functionality within the industry. Consultant Joe Mohen, vice president of Teleprocessing Connection Inc., in Locust Valley, N.Y., reports that almost 80 percent of the large companies he deals with would like to implement ESD to solve administrative problems stemming from the use of PCs and distributed applications. [6]

Echoing the need for an ESD solution is John McConnell, an analyst at Infonetics research, in San Jose, California. McConnell reveals that "a lot of companies are crying for these tools, and IBM and Hewlett Packard will be instrumental in pushing this market forward". [7]

The High Cost of Managing PC Software: A Corporate Asset

PC application software has become a corporate asset. Every day the wheels of Corporate America are methodically turned by individuals empowered with application software such as spreadsheet, word processing, graphics, database, and vertical market capabilities. In today's global marketplace, the cultivation and maintenance of a corporate competitive advantage requires a company to provide its employees with the most productive software tools and utilities available. As with any corporate asset, there are associated costs and concerns. The continued proliferation of new PC application software constitutes a major cost for the initial distribution and installation, with ongoing costs for upgrades and bug fixes.

David Gregson, supervisor of end-user computing at Ropes and Grey, a Boston-based law firm with 850 PCs, is considering upgrading the entire firm to MS-DOS 5.0. However, to do the upgrade, he has to "set aside a full week of 6 to 8 peoples' time". [8] Simple mathematics dictates that such an task is not without significant cost.

More important still is the question of how an organization determines the financial return on such an activity. Joanne Witt of Siemens, who is considering the same upgrade, wonders "how you quantify the value of upgrading to DOS 5.0, companywide. It's very difficult." [9]

An industry-wide perspective offered by a Microcomputer Managers Association Inc. (MMA) white paper recently revalidated (originally issued in 1987), took a hard look at software installation and upgrade costs. The MMA study revealed that to upgrade 100 copies of a typical software package required a total of 300 hours, assuming 4 people work full-time for 9 working days. [10]

Developing an ESD Application is a Non Trivial Task

In 1987, Bank of America developed an electronic software distribution application for internal use to address the above concerns. "Building such an application was non trivial" according to Stanley Pomerantz, vice president and system director for network management systems at Bank of America. Estimated expenses for the ESD application surpassed $10 million. "We did not have any problem, however, when we had to cost-justify our investment", claims Pomerantz. "If every software change costs $50 and we make three changes a year for 15,000 users, then the investment is paid off in less than one year". [11] Although Bank of America was able to remedy their problem in a cost-effective manner, it is obvious that very few companies have the resources and knowledge-base needed to develop an in-house ESD solution.

Software Update Avalanche - With No Relief in Sight!

Some corporate planners cling to the false hope that application software is maturing and stabilizing; therefore, there will be less frequent updates. Ponder on! Corporate planners of this persuasion should take cover, for the exact opposite is already the case. The "upgrade decade" is now upon us. [12] According to William M. Bulkeley, Reporter for the Wall Street Journal, in a recent article entitled, Software Users Are Beginning to Rebel Against the Steady Stream of Upgrades , "the entire software industry has embraced upgrades with a devotion to the 'new and improved' that would shame a soap maker". [13] Support for Bulkeley's position comes from the software industry itself. According to W. E. Peterson, executive vice president of WordPerfect Corp., "you can count on upgrades like the earth revolving or young hearts turning to love in the spring". [14] With upgrades rolling out regularly every 6 to 18 months, the question is no longer if or when, but why?

A handful of answers come to mind. Certainly, additional revenues in the form of an annuity stream tops the list. WordPerfect calculates that upgrades account for 5% to 10% of their revenue most months. [15] Jeffrey Tarter, publisher of "Soft Letter" (a software newsletter), predicts a shift in emphasis from new product rollouts to upgrade strategies, reporting that many software developers in mature markets, "are already generating 20% to 30% of their revenue from upgrade sales." [16]

A second reason for software updates is keeping pace with evolving hardware requirements and advancements. This, coupled with the opportunity to introduce new application functionality, is hard to resist. Another arguable case for application updates is to keep seasonal software synchronized with prevailing laws, such as tax codes that change every year. For the time being it appears that software upgrades have been added to the infamous list of death and taxes!

The Network Advantage

The industry-wide adoption of LAN-based PC connectivity has provided significant growth opportunity for ESD applications. PC systems managers have begun to see substantial cost savings when using ESD functionality on a LAN-based network. "Networking itself makes upgrading simpler in that it offers a central point", says Keith Herron, microcomputer manager for Weil, Cotshal & Manges. Herron reports that software upgrades across the network, "require half the time it takes to upgrade each machine individually." [17]

In contrast to the Travelers Insurance example noted earlier (pre LAN-based activity where 5 engineers spent 2 months updating 350 PCs with a single application upgrade), is their current LAN-based software distribution process, where 3 engineers keep all applications updated on 1600 workstations. [18] Gregson, at Ropes & Gray, is a LAN ESD convert. He is planning to network his company's 850 stand-alone PCs . "One of the main reasons is so that we can put one version out there on the network -- say a WordPerfect upgrade -- so everyone will use the same version. It will be an incredible savings of time." [19] Gregson has in one breath voiced a key benefit of LAN-based ESD software distribution -- as well as raised the more subtle, yet extremely complex and controversial topic of PC software licensing. It is one thing to distribute software efficiently and yet another to maintain compliance with the myriad software copyright laws.

The PC Software Licensing Quagmire

Within the PC software industry today, there are no fewer than 9 variations for administering software licenses (copyright protection), and thus collecting revenues. They are:

* *Workstation license* - one license per installed workstation, independent of user
* *User license* - one license registered to a user, not a physical workstation
* *Group licenses* - specific number of licenses purchased for a workgroup
* *Site license* - unlimited use of software at a specific geographical location
* *Corporate License* - unlimited use of software within a corporation
* *Unlimited License* - unlimited use of software by anyone (e.g. "shareware")
* *Concurrent or dynamic licenses* - maximum number of concurrent users at any given time. When a user is finished running software, application is available to next user.
* *Volume Purchase Agreement* - number of copies and how they are used is negotiated based on the quantity of software purchased over a specified time frame.

* *Negotiated Support Agreement (NSA)* - addresses not only the licensing issue but the whole process of managing PC software. NSA is a tactic that was recommended four years ago by the MMA in their white paper addressing software upgrade issues. An NSA is neither a site license nor a volume purchase agreement , but rather it provides the structure necessary for policies that can simplify the upgrade process, jettison copy protection on both software and documentation, and facilitate the transfer of information among vendors and users on bugs and bug fixes. [20]

With all the confusion surrounding software licensing practices, many companies are delaying their software purchasing decisions. Jane Morrissy, in her article Piracy Wars Obscure Licensing Dilemma, asks the question on the minds of most companies, "how can anybody be expected to sift through - and adhere to - the morass of software-licensing agreements for each vendor's [software] packages?" [21]

The answer echoed in many of the current articles about software licensing complexities, is that "today, PC managers are finding the task of complying with complicated or costly licensing policies burdensome and often a barrier to upgrading".[22] The word "barrier" should be read aloud as "Lost Software Revenue" to all contributing software vendors. Although some vendors are getting the message and beginning to simplify their licensing requirements, most software vendors have not.

What is obviously needed is a more rational, simplified, industry-wide approach to managing application software licensing schemes. However, since the MMA white paper was published 4 years ago covering suggestions such as the negotiated support agreement, there is still a conspicuous absence of an industry-wide approach to this issue. The MMA continues to be very vocal on these topics and they appear to be the driving force behind most of the initiatives under discussion today.

Given the absence of an industry wide software licensing methodology, ESD products that provide a process-related tool set for establishing, verifying, and managing PC software licensing compliance, are in great demand. According to Morrissey, a PC Week analyst, there is a tremendous need for ESD solutions that manage license compliance. She reports that companies are in "desperate need of adequate tools to ease compliance with the myriad software licensing schemes". [23]

The good news is that ESD solutions are rapidly evolving, providing automated processes for authorizing, tracking, and auditing license usage. Companies should begin to look closely at the types of licensing schemes they have inherited via software purchases, and bring their organizations into licensing compliance.

The problems of managing and auditing PC software license compliance will only worsen if ignored.

Software Licensing Compliance - No Longer an Option!

The bottom line, however, is that companies using PC application software can no longer afford to ignore copyright laws. Recently, the Software Publishing Association (SPA) initiated lawsuits against firms ranging from small vertical market resellers such as Easy Dental Computer Systems (sued by Novell Inc. in March 91), to mammoth corporations such as Gelme, Italy's largest distributor of alcohol and soft drinks (tagged in May, 1991, by international SPA affiliate Business Software Alliance). [24]

The SPA is hot on the enforcement trail of network distributed applications that are magnifying the problem of software licensing control and verification. According to R. Duff Thompson, vice president and general counsel for WordPerfect Corp., an SPA member in Orem, Utah, networks are an implementation device that "make piracy easier, thus we see the likelihood of copyright infringement climbing and we're going to attempt to forestall that". [25]

Taking public education of software copyright laws a BIG step forward, is an oversized billboard near the San Francisco airport and 3 other major cities. The billboard features an enormous pair of handcuffs , with the caption, "Copy Software Illegally and Get This Hardware Absolutely Free." [26] Although there is some debate on the methods uses by the SPA to educate via billboard badgering, few can argue with the principle of copyright protection.

When End-Users are Responsible for Software Installations/Updates

In the absence of automated ESD tools or utilities, other problems quickly become visible such as a lack of PC configuration standards. You only have to install one PC application manually to appreciate the frustration experienced by a novice PC user trying to answer a barrage of questions about his or her PC hardware and software configuration during software installation or upgrade. In companies where there is no centralized IS department or PC software administration support, each individual must suffer through the arduous task of "Please place Application Set Up Disk #1 in Drive A:" and "Please indicate which Memory Manager utility you are using," etc.

In addition to the obvious confusion and inefficiencies created by the above scenario, there is the high probability that IF the software is installed, it will not be configured optimally.

An even more worrisome thought involves mission-critical applications. "If a user works with Lotus 1-2-3 [from Lotus Development Corp., Cambridge, Mass.], then it may not be important if he/she has version 2.0 or version 3.0. ", notes Pomerantz. On the other hand, "if he or she uses a mission-critical application and has outdated software, that oversight could cost the company a lot of money". [27]

Standardized, Configured PCs can Still Be Personal!

Additional concerns arise when end-users have sole control of their PC and PC software, such as a lack of any standardized disk/directory organization, poor memory management, complex printer configuration, etc. Each user left to his own means will find a way to create a PC environment different from his co-workers. Although freedom should be given to a user of a *personal* computer, choices about where and how applications are stored and optimally configured should not be left to user discretion. Attempting to update even 20 uniquely configured PCs, can be worse than a nightmarish stroll down Elm Street!

Some companies believe that standardization via Windows will make their lives easier. Stan Levine, vice president of engineering for Automated Designs Systems in Atlanta, reports seeing a sign on the wall of one large company in Los Angeles stating, "If it doesn't run Windows, it doesn't run here." [28] Although the Windows environment is not a panacea for all the issues raised above, it certainly is headed in the right direction. One example would be the consistency Windows achieves by having only one print driver for a network. This helps to simplify application upgrades.

ESD Application and Data Security

Because ESD solutions are implemented using a LAN, there is considerable debate on the issue of application and data security. When PCs are connected to the LAN, some argue that the application/data asset is more vulnerable to activities such as software viruses, sabotage by disgruntled employees, and unauthorized access and updates. Certainly LAN-based connectivity greatly simplifies the physical distribution of software and data. If appropriate caution is taken, LAN-based benefits can be maintained.

When attempting to control and prevent software virus, the same precautions should be taken in both a LAN and stand-alone PC environment. All PC software and data should be tested and verified to be virus-free before it is distributed via an ESD application. If all ESD software is routinely checked, controlling viruses within an organization becomes a more manageable task. The centrally distributed software will be free of any virus. Alternatively, in a stand-alone environment, software that is typically passed via floppy disks may pick up a virus when placed in multiple PCs.

Concerning sabotage or other inadvertent corruption of applications or data, most well designed ESD products are constructed to permit only authorized activities. Typically, some type of security (user-entered passwords or encrypted files) are employed to identify the PC user to the server-based ESD application. Only after designed security requirements are met is the PC allowed to receive application software or data. Most ESD applications allow for read-only access against the server. This prohibits any unauthorized access to the ESD centralized software library. Maintenance and systems administration activities are usually performed using a custom interface that also generally employs the use of password protection.

ESD - A Good Start...

A number of ideas and issues have been discussed in this paper concerning the functionality of ESD. However, it is essential that ESD applications go beyond the simple distribution of PC software over a LAN. Maturing ESD products are beginning to address the broader range of PC Software management issues, including licensing management and auditing, hardware and software asset management, hardware and software PC configuration management, and a reporting capability to keep track of all ESD activities.

ESD licensing management products should be flexible enough to handle a mixture of the 9 different licensing schemes described earlier. Typically, PC applications are bought from different software vendors, and no two vendors seem to follow the same software licensing blueprint. The ESD application should be sophisticated enough to make the PC software available while transparently logging the appropriate information necessary for maintaining compliance with a specific licensing scheme.

Auditing software licensing compliance will follow easily because there is a defined and visible process for controlling and tracking PC software usage based upon a vendor specific licensing prescription.

In order for ESD software to provide for true "push button" software installation and upgrade capabilities, it must have some knowledge of the networked PCs hardware and software configurations. Either the ESD application must accommodate the recording and maintenance of the information, or it must interface with other systems that contain this data (i.e., an asset management system or human resource database). Some of the more sophisticated tool-assisted ESD applications allow the scripting language access to PC hardware and software configuration details. This information can be used to customize (package) the application software in advance for mass distribution, or at download time for a specific PC configuration.

Certainly, a valuable benefit of having a standardized, process-related approach to PC software installation, is the ability to aggregate the detail data and determine how many PCs, with what hardware and software configuration, have what version of what application, in which work group, located where in the company. This type of information is critical when making cost-effective business decisions concerning the purchase, installation, and upgrade of PC application software.

The perspective gained from a company-wide PC application software installation and reporting process, should make managing this corporate asset a bit easier, and certainly more tolerable.

As companies look toward deploying an ESD solution to help resolve the PC application software madness, it would be prudent to consider all the issues raised here as well as keeping abreast of emerging ideas. There are existing ESD products available today that sufficiently address the areas of, software distribution, licensing management, license auditing, PC hardware and software management, and reporting and analysis. Full featured ESD products can replace the PC software management madness with a rational, centralized, manageable, and cost effective process.

References

1. Raths, David, "Software upgrades are more complex than ever", *InfoWorld*, September 30,1991, page S62.
2. Korzeniowski, Paul, and Desmond, John, "And Now The Software Delivers The Software", *Software Magazine*, May 1991, page 63.
3. Busse, Torsten, "LAN managers struggle with updates", *InfoWorld*, December 2, 1991, page 8.
4. "Software upgrades are more complex than ever", page S62.
5. Hindin, Eric, M., "A Simpler Way to Send Software", *Data Communications*, March 1991, page 43.
6. Ibid
7. "LAN managers struggle with updates", page 8.
8. "Software upgrades are more complex than ever", page S62
9. Ibid.
10. Keefe, Patricia, "Users Bemoan Upgrade-itis", *ComputerWorld*, November 12, 1990, page 143.
11. "And Now The Software Delivers The Software", page 76.
12. Keefe, Patricia, "Users seek breathing room in upgrade avalanche", *ComputerWorld*, December 3, 1990, page 56.
13. Bulkeley, William, M., "Software Users Are Beginning to Rebel Against the Steady Stream of Upgrades", *Wall Street Journal*, September 20, 1990, page B4.
14. Ibid.
15. Ibid.
16. "Users Bemoan Upgrade-itis", page 143.
17. "Software upgrades are more complex than ever", page S62.
18. "LAN managers struggle with updates", page 8.
19. "Software upgrades are more complex than ever", page S62.
20. "Users seek breathing room in upgrade avalanche", page 56.
21. Morrissey, Jane, "Piracy Wars Obscure Licensing Dilemma", *PC Week*, May 27, 1991. page 1.
22. Flynn, Laurie. "MMA tries to make software licensing for LAN users easier", *InfoWorld*, September 30, 1991. page S64.
23. Morrissey, Jane, "Piracy Wars Obscure Licensing Dilemma", page 1.
24. "Piracy Wars Obscure Licensing Dilemma", page 6.
25. Ibid.
26. "SPA Issues Graphic Warning", *InfoWorld*, December 2, 1991, page 104.

References

27. "And Now The Software Delivers The Software", page 76.
28. "Software upgrades are more complex than ever", page S62.

Bibliography

Bulkeley, William, M., "Software Users Are Beginning to Rebel Against the Steady Stream of Upgrades", *Wall Street Journal*, September 20, 1990, pages B1, B4.

Busse, Torsten, "LAN managers struggle with updates", *InfoWorld*, December 2, 1991, page 8.

Flynn, Laurie. "MMA tries to make software licensing for LAN users easier", *InfoWorld*, September 30, 1991. page S64.

Hindin, Eric, M., "A Simpler Way to Send Software", *Data Communications*, March 1991, pages 43 - 46.

Keefe, Patricia, "Users Bemoan Upgrade-itis", *ComputerWorld*, November 12, 1990, pages 1, 143.

Keefe, Patricia, "Users seek breathing room in upgrade avalanche", *ComputerWorld*, December 3, 1990, page 56.

Korzeniowski, Paul, and Desmond, John, "And Now The Software Delivers The Software", *Software Magazine*, May 1991, pages 63 - 76.

Morrissey, Jane, "Piracy Wars Obscure Licensing Dilemma", *PC Week*, May 27, 1991. pages 1, 6.

Rafeld, Michael, J., "Managing Networked PCs", *SCRUG Proceedings* , May, 1992, Burbank California, pages 1-10.

Raths, David, "Software upgrades are more complex than ever", *InfoWorld*, September 30,1991, pages S62 - S64.

"SPA Issues Graphic Warning", *InfoWorld*, December 2, 1991, page 104.

PAPER NO.: 4502

TITLE: PC Integration: A First Step Toward Client/Server Computing

AUTHOR: Melodie Sherwood
Hewlett-Packard Co.
c\o Ella Washington
19091 Pruneridge Avenue, M/S 46LG
Cupertino, CA 95014
408-447-1053

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO.: 4503

TITLE: Information Warehousing: Future Directions in Information Access

AUTHOR: David Blakeney
Hewlett-Packard Co.
c/o Ella Washington
19091 Pruneridge Avenue, M/S 46LG
Cupertino, CA 95014
408-447-1053

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

DAVID W. HABERMAN

SPEEDWARE CORPORATION
1150 HANCOCK STREET
SUITE 410
QUINCY, MA 02169

(617) 774-0440

In 1977, when I bought my first PC, PCs were virtually nonexistent in the workplace. Over the past 14 years the PC has changed the way we do business. In the early '80s very few organizations had a plan for implementing PCs. PCs were an oddity that were provided to some accounting people in the back room. There was no plan for integrating PCs into the workplace.

By the late '80s PCs were a major fixture in the work environment. Most of these PCs were purchased one at a time by individuals within the organization. These individuals usually learned to use the PC on their own.

The boom in PCs within the workplace created many problems. Since systems were purchased on an "as needed" basis there was very little coordination of hardware/software platforms. There was almost no training available, and support was not available within the organization. MIS managers found themselves getting support calls on 50 different PC programs. All purchased without MIS involvement, never mind approval.

We have now reached a point where managing PCs within the organization has become a major management issue. This paper will discuss the issues involved. These issues include:

- Who is responsible for PCs within the organization
- Coordination of software purchases
- Coordination of hardware purchases
- Managing repairs
- Managing software upgrades

- Hardware support

- Software support

- Training

I joined my organization in 1987. At that time we had 3 PCs, an HP150, a Compaq portable II, and Panasonic. Like most organizations we had a mixture of hardware and software. We also had no plan to purchase PCs. We basically bought the least expensive we could find. One year, six months, and twenty PCs later we realized we had a problem. We were not taking advantage of economies of scale. We were not making bulk purchases. Supporting 3 different word processing packages was impossible. Coordinating repairs and warranty work between vendors was a nightmare.

As resident "expert" I was coordinating these tasks on an as needed basis. At one point managing PCs and providing support was taking almost 50% of my time. At this point we put together a PC strategy and long range plan. Since then I have discussed the issues involved with many MIS shops and learned I was not alone.

Who is responsible?

Who in your organization is responsible for PCs? In asking this question I have received many different responses. They include:

Individual department heads - PCs are generally purchased and approved at the department level. Therefore, all coordination of PCs is done by the individual department heads.

Purchasing - Purchasing is responsible for all repairs and purchases.

MIS - PCs are computers.

Some guy in accounting - This guy is the resident PC expert who everybody comes to for advice.

Today more and more organizations are centralizing PC responsibility on a PC coordinator. This person's job description usually includes:

Developing hardware and software standards

Coordinating training

Providing software support

Coordinating hardware/software purchases and repairs

At what point do you need a PC coordinator? In tracking my hours spent supporting PCs, I found I spent between 30 and 40 minutes a week for each PC in PC related tasks. At somewhere between 40 - 60 PCs this becomes a full time position. In speaking with PC coordinators I have found that one person can provide support for approximately 80 - 100 PCs at that point the level of support begins to drop drastically.

Coordinating Software Purchases

In looking through many organizations you can find at least four different word processors, 2 different spreadsheets, and all sorts of database and utilities being used. It is important for an organization to develop software standards. An organization with a standard set of software tools can reduce the purchase costs of their software. Most software companies offer site licenses or bulk discounts.

Standardizing on software allows the organization to maximize its training and support resources. Training classes can be developed and tailored to the organization. Support staff can be trained in a minimum of applications. This will reduce the training expense and allow support people the opportunity to gain greater expertise in the chosen platforms.

By standardizing on software you are better prepared to meet future technologies. Imagine a technology that obsoletes all your current software. If everyone is standardized data conversion is easily managed.

In many organizations you will find many illegal copies of software. This is a dangerous practice. An article in PC Week described a company that paid $70,000 dollars in fines for illegal software. Some disgruntled employees of the company reported the violations to the Software Publishers Association. The SFA came in with federal marshalls and audited the companies' systems. They found over 200 illegal versions of software on the systems. The company paid a $70,000 settlement and then had to purchase the software.

You may think 200 versions of illegal software is a lot. If you have 40 PCs and each has 5 illegal programs on it you've got 200 illegal programs.

The PC coordinator must maintain proof of all software purchases and register all software with the vendors. Software companies are constantly upgrading their products. If you register your software you can usually receive the upgraded versions at a fraction of the list price. If you standardize your software and hardware environments you can develop batch programs to walk your users through the software installations and upgrades. This saves the organization time and money.

Coordinating Hardware Purchases

I have heard many times that all PCs are the same. Anybody who owns three different brands of PCs knows this is not true. MS-DOS may be MS-DOS but that does not account for how the PC vendor chose to manage memory. Many vendors build programs into chips. The system set up programs differ PC to PC. When inserting add on boards many systems have internal switches that have to be set.

By standardizing on one hardware platform you can cut down on troubleshooting time. You can also set up memory management and menu batch programs that are common to all systems. This will reduce the time in involved in setting up PCs for end users.

Standardization will also let you define how you will use external devices. Ever try defining a new printer for a program in windows. Now assume that everyone in the organization has a different printer. This can easily involve days of set up time. Simple issues such as parallel or serial interface can amount to a lot of time if you are managing even 20 PCs.

By standardizing your hardware platforms you can also take advantage of bulk discounts. Even if you don't purchase all your systems at once, many vendors will allow you to spread your purchases over a period of time and still take advantage of discount pricing.

Managing Repairs

I am always asked whether or not I recommend extended warranties for PCs. I have not personally found them valuable. Most PCs come with a 90 day warranty. The extended warranty is usually purchased at the same time. For at least 90 days you have duplicate coverage. PCs are all solid state electronics. My experience has been that if there is a problem it will occur within the first few months.

A support contract for a PC is about \$200 - \$250 a year. A new PC is about \$2000 - \$2500. If you have 10 PCs and have 1 breakdown each year you can afford to replace the PC.

When you bring a PC in to repair they do not "repair" the part. They replace it. PCs today are so modular in nature that it is possible to cannibalize old PCs for spare parts or buy replacement parts yourself. A well trained PC person can manage this without a great deal of expensive technical training.

Software Support

PC software support is usually a big pain in the neck for MIS. PC users assume that if you program in RPG you must be an expert at Excel. Nothing can be further from the truth. To provide technical support on PC software you have to be trained in each individual package. I am not saying that a full scale vendor training program must be in place. I am saying that the support person must learn each individual package either on their own or through some training course.

Many organizations say they do not need a PC support person. They let the support be performed by department "experts". This is usually a person within the department who is familiar with the package.

What most organizations have, is no conception of the amount of time spent on this type of support. This greatly reduces the productivity of the "expert". They can spend a great deal of time trying to solve a problem they are not qualified to solve. This means that the person with the problem and the expert are both unproductive until the problem is solved.

I challenge any organization using this method to ask your resident experts to keep a log on their desk. Ask the expert to record any time spent on PC support. You will be surprised by the results.

Another method I have seen is to let MIS people take turns at providing PC support. Each programmer takes a turn at the support desk and they rotate on an ongoing basis. The problem here is continuity. If there is a recurring question or problem each individual may solve the problem over and over again. A single support person can recognize patterns and develop global solutions, rather than dealing with the same problem 100 times.

I believe that by evaluating the amount of time spent on purchasing, supporting, and managing PCs within the organization that PC management should be coordinated at a central point throughout the organization.

Old MacDonald had a network

Anthony C. Furnivall

SDL/Software, Inc

P.O. Box 601

BUFFALO NY 14213

When first confronting the world of networks, the stoutest heart trembles at the jargon that is used so freely. This presentation will demonstrate how all the talk about protocol stacks, layered architecture and so on really does have a useful part to play in getting the job done. It involves members of the audience and demonstrates through such participation just how to make sense of the barnyard babble!

INTRODUCTION

I can imagine few things more daunting than walking into a networking show, with little or no idea of what the vendors and attendees are talking about. Where else can you hear such a density of meaningless TLAs (three letter acronyms) flying around with such abandon? Trying to make sense of the jargon is a tough prospect. My goal in this paper is to provide something that you can read and use as a reference either before or after (or even in stead of!) attending the actual presentation.

It has been said that the mind retains 10% of what it hears, 20% of what it sees and 50% of what it does, and for this reason the main thrust of the paper is a participatory experience. However, not every one can attend, and even for those who do, some follow-up might be useful.

What follows, then, is a plain and easy introduction to the idea of network layers, and network protocols. It is not intended to be a reference on the subject - indeed, there are several places where accuracy has been sacrificed for the purpose of clarity. However, it should stimulate you into going on and doing more reading.

NETWORK LAYERS

It is almost inconceivable that at some point during the past 12 months, you have not seen a diagram like Fig. 1. This diagram shows what is known as the ISO OSI model. I prefer to think of it as 7-layer network cake.

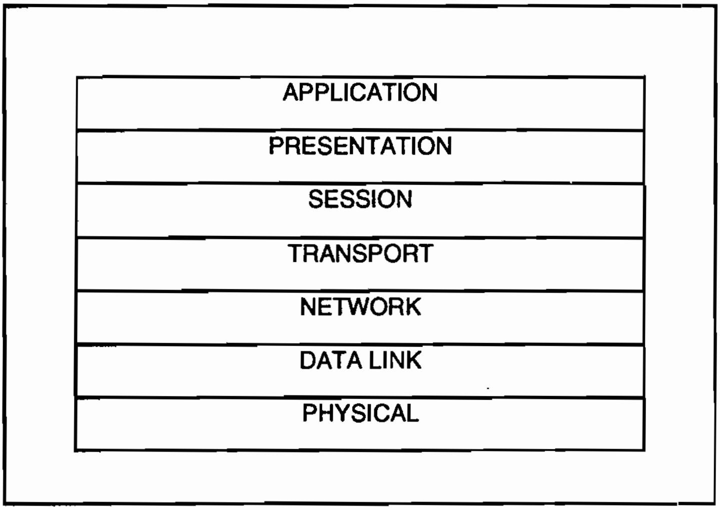


Fig. 1 - 7-layer network cake

This innocent little picture conceals a wealth of information, and a rich potential for confusion. The next few paragraphs will attempt to give a simple idea of what the various layers do.

Why do we need layers?

The first, and most important question to be answered is, Why do we need layers? What is it that is special about the layers in a picture like *Fig. 1* ?

The answer lies in the fact that it is easier to divide a complex task up into small pieces, and then to manage the small pieces individually. This is a common feature of software engineering, people management or almost any other field of human endeavour. No surprise, then, that the same thing holds true with computer networks, or data communications.

To define the entire set of tasks in one complex document would make it almost impossible to connect any computers together. What has happened, therefore, is that the task has been broken down, into reasonably logical pieces. These pieces represent the layers of the International Stack Of Protocols.

Physical

The physical layer is concerned with how the data is represented in terms of voltages and frequencies. It also deals with the size and shape of the connectors used to plug in to the network, and the types of cable used to carry the signals. Pretty boring stuff, for the most part, but very important. If you have ever tried to plug a three-pin plug into a 2-pin socket, you can imagine how frustrating it is not to have the right connections for a network system. Especially since the little adapter that you get for 79¢ at K-Mart costs several thousand dollars in the network store!

What we have, then, at this layer is the ability to create a 0 or a 1, several times a second (actually, several million times a second, but what's a few orders of magnitude between friends).

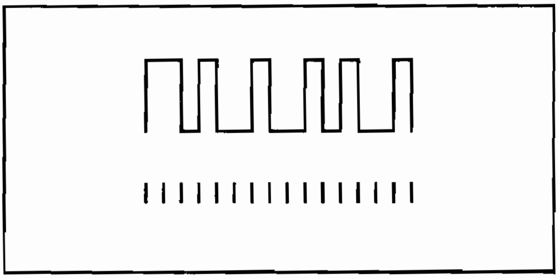


Fig. 2 - Signals varying over time

Data link

The data link layer is concerned with how the signals of the physical layer (the '0's and '1's) change over time. A voltage change is used to represent a data value. You may like to think of the options available at this level rather like the difference between AM and FM radio.Both types of radio are useful for bringing information into the home, but they are not directly compatible.

Some of the buzz words that you will hear in connection with this layer are terms like Ethernet, Token-Ring and MAC layer. We will explore Ethernet and Token-Ring a little later. MAC stands for Media Access Control, and refers to a common standard for moving data out onto the cables of the physical layer.

A sequence of bits, at this layer is called a frame. The bits have specific meanings, and these meanings are described by the various protocols which operate at this layer (such as Ethernet and Token-Ring). In addition, the bits are grouped together, generally in groups of 8. However, reflecting the truly catholic background of networking, these groups of 8 bits are called not a byte, but an octet. There is a good reason for this, but it can be confusing for beginners, especially if you have an interest in music, and regard an octet as a piece of music for 8 players!

```
10101011010110110101011010000101110101101101011010101101

101010110|1011011|1010110|10|0000101|1010110|1010110|0101101|
```

Fig. 3 - A septet of octets (56 Kinda Binary Signals)

Network

The network layer is concerned with making certain that the data is delivered to the right place. This is rather like making certain that you get your mail, and no-one else's. There are several interesting features of this analogy.

Just as much of your mail is junk-mail, addressed to 'OCCUPANT', so much of the data on the network is potentially junk-mail. It actually has rather more value to someone, but a lot of the messages on the network are addressed to everyone, in the hope that someone will respond.

The second interesting feature concerns the way in which the mail (read network messages) gets delivered. Rather like the Post Office, there is no guarantee that the mail will ever get through, or if it does, that it will get through in the right sequence, or if it does that it will all get through. This is rather like ordering a new set of tools from Sears, and receiving the bill before the tools!

Thirdly, just as in the mail system, the data is now called a packet. (What started off as a bit, in the physical layer, and became a frame in the data link layer has now changed its name yet again. However, we are still talking about bytes (or octets).). Like most pieces of mail it has an address to which it is (hopefully) delivered, and a return address from which it came.

Finally, just as much mail is local in nature, a large amount of it travels to different ZIP codes, or different states, or even different countries. The address on the packet indicates exactly where it has to go, but different parts of the mail system read and use different portions of the address information.

The network layer is like the infra-structure of the postal system. It takes care of finding the destination for the packet in question, and finding the best way to deliver it. However, it does *not* represent or include the additional intelligence that users of the Postal System need. This intelligence is contained in the next layer.

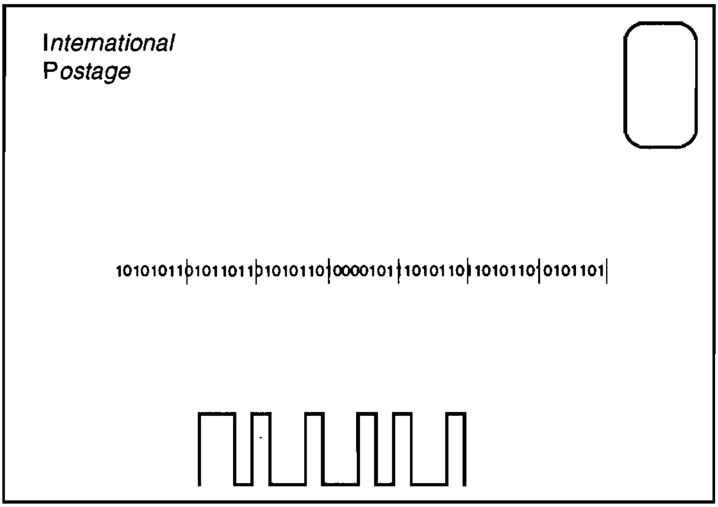


Fig. 4 - A simple network packet – *Truly Child's Play*

Transport

At this point, I have to make a confession.

On various occasions in the past, I have been late making a payment on a credit-card. What happened was that the credit-card company sent me a bill, and I guess it got mislaid, or otherwise ignored. In their kindness they sent me another, and another. Pretty soon, though, they resorted to a different kind of network system, and started calling me at all sorts of hours of the day and night!

What they were doing, in effect, was adding just that intelligence that is missing at the network layer. They had a reasonable expectation that the message they sent me (the bill), would result in a response from me (also via the mail). When this did not happen, they started off by sending out a copy of the bill. Eventually they realized that nothing was going to happen, and changed their tactics.

This is the function of the transport layer. It exists to guarantee a reliable, correctly sequenced stream of data between one node on the network, and another. It includes provisions to keep track of the number of packets in the stream at any time, and to let the sender know if any of them were received out of sequence. (Remember those tools from Sears).

Session

:HELLO!

What happened to our easy introduction to networking protocols. If you glance back again at Fig. 4, it doesn't seem a particularly simple procedure to understand so far. The fact is, that in terms of the plain old terminal systems that we all know and love, we have only created the equivalent of a serial cable between an ATP port, and a VDT. (Watch out for those acronyms - I should have said between an Advanced Terminal Controller, and a Video Display Terminal). The reason for this extra complexity is because rather than having a fixed, permanent, dedicated circuit between the terminal and the computer, we have an open, shared, virtual circuit connecting them, and all the lowest layers are concerned with is managing that virtual circuit.

The :JOB of the session layer is to begin the process of interacting with the computer at the end of the cable. MPE provides session layers services every time you log on, or every time you stream a job. Session layer services are concerned with providing access to data in a controlled way, with accountability. These services include file security, resource accounting and so on.

This is generally the point where you begin to heave a sigh of relief, and begin to have something concrete to which to relate. Indeed, the top three layers are the easiest to understand. However, it helps to go through them to get a sense of how networking can complicate things.

On a VAX system, for example, you don't say :HELLO, you say $LOGON. I have sat in front of a VAX system and tried to persuade it to respond to MPE commands, with very limited success. Fortunately HELP was one of my few successes! When we are tying together different computers with a network, some provision has to be made for a common sequence of commands to gain access to these session services.

Finding out the names of all the files available to you, for example, might involve using any or all of the following commands:

- LISTF
- FSTAT
- DIR

The session layer is responsible for making access to the computer as easy as possible.

Presentation

If we type LISTF on an HP system, we expect to get a list of information about a set of files. However, this information is not stored directly in the same way that it is presented to us. What we have is a software routine (think 'layer') which takes data in one format, and transforms it into another, for presentation to the user. (Incidentally, if you think that a user is soon going to be called an application, you're right!).

Other presentation services include such arcane things as making certain that the bytes which make up a 2 or 4 byte structure are presented in the right sequence, and making certain that the data structures which are expected at a certain point are indeed that data structures that are provided. The problem of byte sequencing is also known as byte-gender, which just goes to show that sex rears its ugly head in all sorts of strange places!

This is very important, however, because if I have a value such as 123456 on an HP machine, it could become any of the following, depending on the byte gender of the machines involved:

-499122175

1088553216

16793826

As you can imagine, transferring funds from a bank which runs HP to a bank which runs, say, Tandem equipment could result in some very strange accounting unless the presentation layer is firmly in place. The same is true, incidentally, for other very slow networks, such as magnetic tape.

I recently completed some data conversion for a client where they were moving data between two systems, and one of the was a VAX. I was doing the conversion on an HP, and sent of a tape with all ASCII data (I was smart enough to avoid problems with binary formats, because of the byte gender differences), but we discovered that the HP and VAX systems represented signed ASCII digits differently. This was my first (and hopefully last) introduction to the Signed Separate clause in COBOL!

Application

The application layer is the layer which provides real services and results to real users. Sometimes this can be an interactive user, and sometimes it can be just a program somewhere which wants to reach out and touch someone one the network. Among the sort of things which you can find happening at this layer are file transfer, terminal connection and network management.

Once we reach the Session layer, the services that are provided are very tightly interlinked. This is in contrast to the lower four layers, where it is quite possible to mix and match. The freedom which this leads to can be confusing because it is important to make certain that both ends of a conversation are speaking the same sort of language.

PROTOCOLS

Our brief and somewhat unguided tour through the 7-layer network cake has touched only briefly on what the various layers accomplish. The purpose of protocols is to determine *how* they accomplish it.

A protocol is defined as:

- The customs and regulations dealing with diplomatic formality, precedence and etiquette.

What we're going to do next is to take a quick look at some of the protocols which are popular at each of the three lowest levels.

Physical layer protocols

At the lowest layer, the physical layer, there is not really much that can be called a protocol. The way in which most networks get data on and off a piece of wire is a rather strange technique called Differential Manchester Encoding. All you really need to know about this is that it works, and provides several additional benefits for the guys next up the line, the data link layer.

Data link layer protocols

The big players in the data-link league are Ethernet, along with its cousin IEE802.3, and the blue-flavoured Token Ring. This is where the first of the crusades is waged, with people arguing in favour of one approach, and other people taking a contrary point of view.

Ethernet and IEEE802.3 are virtually identical. Unfortunately, from a genetic point of view, so are a man and a dog. In both cases, the species can co-exist but communication between them is of limited functionality. Ethernet also has a poor foster sister called Localtalk, which is used to tie together Apple Macintosh computers, and is based on the same concepts as Ethernet. These concepts are known as Carrier Sensing with Multiple Access and Collision Detection. This mouthful is more frequently known as CSMA/CD, and is a notable departure from the rule that complicated networking terms have three-letter names.

What happens here is that anyone can talk on the network at anytime (Multiple Access), just so long as no-one else is (Carrier Sensing). When they are talking on the network, they keep listening to see if anyone else started around about the same time (Collision Detection). If so, both parties stop, and wait a bit before trying again.

This is much easier to demonstrate than to describe, and is the techniques used by most people in most walks of life, most of the time. (Yes, there are exceptions, in the computer world they are referred to as jabbering nodes, in the real world they are called boors).

In contrast to CSMA/CD, the other common way of controlling access to the network, is to pass the pipe of peace. This is like the way in which speakers in Congress are scheduled (I never promised you a reality based example!), and guarantees that there is an orderly progression of opportunities to speak. Each node on the network gets a special token, and can only access the network when it has the token.

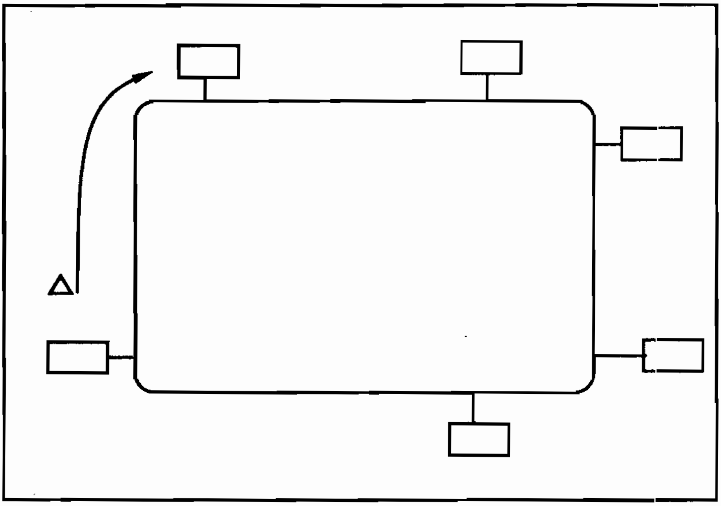


Fig. 5 - One-way token rotation

From a technical perspective, Ethernet is a non-deterministic protocol, and Token Ring is a deterministic protocol. This means that we can not predict who will talk next with Ethernet, but that no-one is unduly delayed, while we always know who is going to have the token next, but we may have to wait for it to go all the way around the ring, if we are busy.

Probably only the Super Bowl or the World Series generate more heated discussions and arguments about the merits of opposing teams than these protocols. The truth is that both work, and work well. Either network protocol does the job, but anyone who wants to, can ring the changes on the old arguments. Most people just get on with the job of implementing one or the other.

Network layer protocols

Network layer protocols, you remember, are concerned with making certain that the correct recipient of a packet can be identified. This involves not only tracking them down, but making certain that you can remember where to find them later on!

The most common protocol in use at this layer, is a protocol called the Internet Protocol, which is almost always seen with its buddy, the Transmission Control Protocol. These protocols were developed for use in the Department of Defense Advanced Research Projects Agency (ARPA) network, and have become a de facto standard. Everyone claims to be able to deliver a TCP/IP implementation, and this is one of the easiest ways to hook up different computers.

Now that the International Standards Organisation (ISO) has gotten in on the act, there are several new protocols available which match more closely the other standards being promoted by ISO. All of these protocols, however, serve the same basic purpose, that of making certain that the mail does indeed get through.

SUMMARY

Networking is a very complex subject - there is no doubt at all about it. However, it need not be unnecessarily off-putting. While some of the concepts are strange at first, they are also self-evidently useful, and serve clearly identifiable purposes. The best way to become more familiar with the language and environment of networking, is simply to do it. Attend training classes, read as much as you can, and put your learning into practice as much as possible.

There are several opportunities to become more familiar with networking, and while some of them seem to exclude beginners, almost all of them have something of value to people with varying degrees of experience.

5003

UNATTENDED DATA CENTERS...
FANTASY OR REALITY?

Frank J. Pinkela

Unison Software, Inc.

408-245-3000

• ABSTRACT

Today's modern data center, while boasting the latest in hardware technology, continues to struggle with the implementation of software that can automate the operational daily production activities. While much has been written on "unattended" operations and "lights out" data centers how much progress has really been made in actually accomplishing these objectives?

Does top management understand the cost trade-offs associated with the data center in terms of personnel turnover, production reruns, policies and procedures regarding production schedules, user submitted jobs, backups, restores, report distribution, and console messages. How will today's data center manager deal with managements request for more productivity and less people? What will be the affect on the data center with the increased use of UNIX boxes? Is it possible to continue to increase these "operatorless" tasks through software?

This presentation will focus on industry trends and case studies in the move toward more automated data center activities. Included will be data from both HP and IBM data centers as well as information from vendors and consultants.

This talk is directed toward supervisors and managers of data center operations for all HP platforms, MPE V, MPE XL, and HP-UX.

• BACKGROUND

A recent gee-whiz article appeared in an AFCOM publication (Dec. 1990) stating - "The National Commission on Superconductivity recently recommended developing a super computer operating at "petaflops" speed, about one-million times faster than today's fastest computer. By contrast, it is estimated that the human brain computes in the 10 "teraops" range, faster by a factor of 10,000 than today's best super computer."

Figure 1 shows the current updated offerings from HP in terms of relative performance of the HPPA machines. Please note that the current upper end machine operates at about 50 MIPS and costs about $1 million. That makes you (or more specifically your brain) worth, on the above scale roughly **$20 billion.** Amazing isn't it!!

This anecdotal reference clearly points to the continue advancement in the computing power of hardware as we progress toward more and more efficient ways to process data in both our personal and business environments.

The objective function in automating the data center should be in maximizing efficiencies and minimizing costs. Look at what has happened in your own company's business operations, automated production lines, robotics, automated warehousing and inventories, automated cash registers, point-of-sales equipment, CASE tools for development, computerized-digitized everything. Has automation of the computer data center kept pace with these advancements? Part of the problem, I think, is the failure of the "keepers" of the automation processes to use these tools in their own environment. Kind of like the old bromides about the shoemaker's son going barefoot or the doctor unable to deal with his own prescription for good health.

•TRENDS

••SMR STUDY RESULTS - by Sentry Market Research

These study results address a wide range of automation issues and measures the current impact of automation at a wide variety of computer sites. Typical respondents were involved in technical operations & support and systems administration.

•Research shows that AO tools are delivering results on target with customer expectations and further suggest that employee morale has NOT been negatively affected by the automation of the data center.

•A sampling of the group showed they made a significant investment in documenting the "before and after" effects of the automation process.

•Cost savings were tops on the justification list.

•Head count reduction was the least important item on the list.

While the market place abounds with automation tools, most vendor products fall short of user expectations for interchangeability. A study I conducted last year with 10 HP data center managers indicated they were looking for more "single-vendor" solutions and a move toward "open" exchange of information, common operator interfaces, and alliances amongst vendors to standardize automation approaches. Additional concerns were for the forthcoming "mixed" operating system platform, viz. MPE and UX, that will become evident in many shops in the next 12-18 months. How will your current data center procedures and software operate in this mixed environment, if at all?

The minority of data centers have "lights out" as their ultimate goal. This is as much a testimony to the current state of AO technology as it is of the relative uneasiness users have with totally unattended operations. The level of comfort seems to be around more of a "dim the lights" rather than "lights out".

•OPEN ARCHITECTURE FOR DATA CENTERS
(Ref Figure 11, AUTOMATED OPERATIONS -The Wheel)

•CHALLENGES FOR DATA CENTER MANAGERS
-In proceeding toward AO, DC managers had the following insights,

A. CRITICAL SUCCESS FACTORS

Managers need to have a handle on "critical success factors" in order to establish a baseline operational position and to measure the effects on additional workloads. Management is now asking, and will continue to ask the question, "How can we do more with less resources?" Are you prepared to answer , objectively?

The term "critical success factor" is used to describe the various production activities and performance factors that highlight the productivity and efficiency of a data center needed to communicate with top management as to operational performance data...how are current resources being used; why are additional resources needed; and, what is the benefit to the enterprise if the capital is directed to this project; are there better projects than this one for this capital; etc. Some of these measurements might be:

-resource usage - cpu, disk, tape, network, supplies.

-user computing - planned, unplanned, peaks, need vs want.

- on-line response time - demand periods, perception vs reality.

-system availability - downtime, weak link in availability.

-batch job throughput - number of jobs processed.

-problem analysis - job failures, system downtime, prod reruns, scheduling bottlenecks.

•SUMMARY

Yes, unattended operations are a "reality."

Many are moving into more and more automation of the data center for reasons of economy, available resources, ability to respond to situations, and the <u>need</u> to do more. Some have arrived and are fully committed to this automation concept, others are making do, and still there are a few that have no interest. If you're in any of the above, except for the last category, then the following may be useful to your quest for automated operations.

1. <u>Evaluate</u> your in-house software for scheduling, restart, backups, report distribution, and console management. Are your packages working together or is it feasible to make them work together in the future. ESA...if you can't <u>e</u>liminate it, then <u>s</u>implify it; if you can't simplify it; then <u>a</u>utomate.

2. <u>Differentiate</u> between your current costs and what will it cost to support additional planned workload. (baseline budget).

3. <u>Consider</u> expert advice. Make use of your fellow users, companies that have advanced the state of automated operations.

4. Evaluate current software packages. Most products can get the job done. Question - how easy and effective to implement. Not only your current "pains," but those you anticipate and those that others have experienced in the process.

5. Document your plan. Plans alone don't mean a lot, however, *planning* is everything. Identify the costs and savings that will be realized through the purchase of new software and track your progress (both successes and failures.) **Remember, you may have to do it again.**

While the cost of upgrading can be expensive, the payoff will be efficiently run and competitive data centers.

Paper#: 5004

Title: Automating The System Manager/Operator Environment

Author Name: Nicholas Powell

Company: Comprehensive Software Solutions
300 N. State, Ste 3125
Chicago, Il 60610
(312) 222-0758

Objective:

This talk is intended to appeal to anyone in MIS who spends a large portion of their time dealing with the needs and demands of end users. This includes System Managers and Operators. In a small MIS department the MIS manager, programmer/analyst, System Manager and Operator could well be one person; whereas, in a larger organization, each role could be filled by a different person.

The role of an MIS department should be to provide reliable, high performance, timely, and cost effective information systems for the users. In reality, most shops spend much of their time trying to stay on top of the day-to-day requirements of their existing systems, thus not allowing much in the way of resources for meeting the changing needs of the users and management.

The purpose of this paper is to provide ideas for ways to automate some of the day-to-day tasks that system managers, operators and MIS departments have to face, thus giving them more resources for meeting the above mentioned needs of the users.

By treating the needs of the MIS department in the same way that you would treat the needs of another department, you can eliminate waste of time and resources, and provide a communication within your own department.

In other words, just as you would design a system for your users, so you should design internal systems the same way: that is to say, Requirements Analysis, System Analysis, Design, Development and Maintenance. In fact, since developing systems using these steps is your business, so you should track the resources and time involved in the development cycle using internal systems developed using those same techniques! Recursive internal systems, no less!

By doing this, you will make better use of your time, and be able spend more time being more responsive to the needs of the users. In this you will have the time to 'be Proactive, not Reactive'.

1) Monitoring resource usage

This can apply equally well to human as well as computer resources. Most MIS departments manage both types by word of mouth or if you are very lucky, a memo. Some shops use electronic mail to pass on requests for resources as well as to maintain resource usage schedules, really just an electronic memo and bulletin system.

A better solution is to maintain a project and resource tracking system that can manages all the internal systems that are currently in development.

As well as tracking MIS projects, the system should be able to track any type of resource, including any inhouse tasks that will require significant investment of time and effort. It should be able to account for time spent on a project, the schedules for the project and the individual resources, the cost of developing the system for that cost center, and how far through the development cycle the project is.

By using such a tool, you will be able to generate task lists for all the staff involved in any given activity. This will also allow you to track the current status of all tasks.

You will be able to track how closely the estimates of the duration of the task match the actual time spent on it. This ability to review estimates of the duration of projects will allow you to refine future estimates, and improve the way your time is scheduled.

An example of such a tracking system can be seen on the projector.

If a project, task or activity tends to recur, the indication is that it should probably be automated! If it is repetitive then you are probably wasting resources doing it manually! And in fact, most of the activities below fall into this category.

2) Data File Capacity Management

How many times have you had a dataset fill to capacity on a production database? If it has happened once, that is once too often!

By automating capacity management for all databases and data files you save the person responsible from many manual trips into query or the database utility. On these occasions, a dataset might be missed, or the checkup might be forgotten due to time pressure.

A suitable alternative would be to have a database containing the names of all databases and data files, with a flag indicating whether it is an Image file or an MPE or KSAM file. A program would then be able to read the dataset indicating which databases should be checked, and create a custom Job Stream set up to run one of the standard capacity utilities, eg: DBFULL, against each entry.

Such a database capacity monitor can easily have it's output redirected to disc (using ;STDLIST=filename). A short utility program then runs to analyze this information, searching for any record where a capacity is greater than, say, 80 percent. This utility can then, as an example, create an HPMAIL message with the name of the data file, alerting the appropriate person of an impending capacity problem.

By this means, only an exception list will ever be created; thus eliminating stacks of printed output requiring manual attention, and avoiding the likelihood that file capacities will not be checked in a timely manner.

3) Shutting down and restarting the system

A good use of automation is in standardizing the steps that are executed when system events take place. In shutting down or starting up the system, a standard job can reliably carry out all the steps that are usually taken manually. Many of these are steps that cannot be performed easily using SYSSTART or command files.

For shutting down the system all of the following can be integrated into one: warning the users to sign off, setting the limits down, down'ing modems, shutting down HPMAIL and the HP SCOPE collector, signing non-console users off, saving spoolfiles off to tape, etc.

For re-starting the system, the inverse of all of the above can be executed, including starting new log files, running HP's PREDICTIVE support, making special changes to the spooler queues, running HPMAIL's maintenance job, etc.

One main issue in having a master stream running other jobs, is that of synchronizing the execution sequence of events. An example of a good solution to this problem is employed by HPMAIL, which writes a record indicating that a process is executing when it starts, and when the process finishes the flag is cleared. The master process, by looping and checking the status of the son process, can tell when it is time to proceed.

By automating this step, you will bring your system down (and up) in a consistent manner every time, and with only a few keystrokes.

The users will come to know exactly what to expect when they receive a message telling them that the system is going down in five minutes. They will know that they have to save their work, and that the time to shutdown is not a 'flexible' 5 minutes (in other words half an hour!).

4) Batch management and scheduling

Surely all MIS departments have reports in the following categories: run on user demand, and run nightly, weekly and monthly.

For the regularly scheduled reports, most of the time the same programs will be run in the same order. As circumstances vary, some reports may be run multiple times per night and at other times a report will not be required to run at all. Sometimes a different program will be substituted to perform a variation on the usual task.

Rather than maintaining many Job Stream files with cryptic eight character file names, and multiple copies of each - all subtly different, a better way would be to create a Job Scheduling database.

Each Job Stream would require a master entry with the name of the stream. Also of value would be the frequency of the run, the approximate duration, and the standard time the run should be scheduled at. Also the sign-on and INPRI and output priority. There should also be the capability to enter a meaningful description of the function of this run. The capability should exist to copy a Job Stream entry (if the new variation will be reused frequently). Attached to this entry will be line items containing the names of programs to be run or additional jobs to be streamed. By having a text entry screen hanging off each entry, MPE commands to be appended after the run statement can be entered.

At the beginning of the nightly run, the operator launches a utility program to check through the database, searching for jobs that should be streamed. This would then build an appropriately structured job and stream it.

The advantages of this approach are that: all jobs will be run at the correctly scheduled time without manual intervention, the format of all job streams are standardized, each job can have a meaningful description associated with it, any standard TELLOP messages indicating the job's progress can be generated automatically, making changes to a JCL is not going to cause an abort because of a typing error, and changing the sequence of execution of programs in a stream or adding extra steps does not require skill with MPE or an editor.

A similar approach can be taken for reports run on demand by the user. Rather than just letting the user stream the job with no regard for the current system load or any rush jobs that might be executing, it is better to set up a user report request system.

In such a system, each entry by the user represents a run that they want executed. By providing a lookup table keyed by the current user name, that user can only request reports that they are allowed access to. Each request is assigned a unique id number and the date and time entered is logged. User comments can be associated with each entry as instructions for the operator.

The operator would be able to check this system without having to run the full menu to see if there are any requests pending (rather like the NEWMAIL command in HPMAIL). If there are outstanding requests, the operator can run the request system and examine each open entry. If he chooses to run the report immediately, he can simply press a function key and the job will be streamed, and the entry will be logged to show the date and time that the job was launched. If desired, the operator can leave a message for the user indicating the why a report can not be run, or when it will be run.

5) Expediting user report requests

MIS developed or end user developed reports? The work load of the average department means that some user requests won't be expedited in a timely fashion. By means of an end user report writer, while retaining overall control, the load is taken off the MIS department, giving the users fast and meaningful results and improving morale.

By giving the user an understanding of the processes involved in developing reports, they will have a common ground with MIS in discussing other aspects of the system. Rather than treating MIS as a black box, where requests go in and reports come out (if you are lucky), users will feel that you are being responsive to their needs.

The criteria for separating reports into MIS vs end user responsibility, include the complexity, whether files are updated, and interaction with other system modules.

User request tracking system: By creating such a system, MIS can manage the incoming requests. Those that have to be carried out by MIS can be prioritized, and logged as they are dealt with.

This tracking system would have to be keyed by product, version number, and internal request number. It should include information such as; the user priority, the MIS priority assigned, a request description, the current status of the request, the date requested and by whom, the resource charged with handling the request, a solution description, the date completed, the version the request is completed in, and a reason if MIS is unable to complete the request.

An example of such a system can be seen on the projector.

The user should be kept informed of the status of their requests by printing out a log regularly. This will prevent the frustration of the scenario where the user has to keep asking what is happening with their request, and the MIS representative has to keep saying that they don't know; until either the user gives up, or the MIS person does the report just to get the user off their back.

6) Quality controlling 3rd party or internal MIS products

The golden rule is to thoroughly unit, system and integration test every module that will be used on the system.

This may seem excessive, and the temptation is to just go into production and wait for feedback; but as an analogy, how would you feel if the engineers at Boeing, McDonnell-Douglas or AirBus Industries designed and built planes you have flown in the same way? Just build it first, put passengers in it and then see if it will fly! The time spent testing will save you from hundreds of man-hours of 'fire-fighting'.

Setting up the environment: Always create a test account away from live system, totally self-contained. It should mirror the live environment, except that security and low capabilities should be applied, so that if a program tries to access files outside the account, you will quickly discover it.

Timing and re-creating test environment: To allow you to easily recreate the same test run, use 'bench' files for tests on screens. For reports, use an FCOPY to a log file at the beginning of the job with the date, time, module name, etc (using :EOD to end it) and FCOPY with ;ACC=APPEND to the same file at the :EOJ with the same information.

Testing comparisons: Parallel testing & comparison of results and run times should be logged to a database, keyed by 3rd party or internal product name, version number, date and test number. Include as line items the module tested and by whom, the number of users, the elapsed time and cpu time, any errors encountered, and the problem number to link into below problem tracking system (if applicable).

Problem logging system: By setting up a such a database, MIS can track any problems reported and manage bug fixes (as opposed to enhancements).

In designing such a system, you will need to key the information by 3rd party or internal product name, version number, internal problem number, and 3rd party call number.

It should include the problem description, the call status, the date discovered and by whom, the date reported, the person who took call, the solution description, and the date and version when the problem was fixed.

An example of such a system can be seen on the projector.

This internal system should include the ability to print out fix logs and outstanding problem logs, by any of above criteria.

Conclusion:

It may seem like a lot of internal systems to develop simply to track the flow of user and MIS requests and problems. By developing the systems in-house using a 4GL, by purchasing a 3rd party products, or by developing / purchasing individual modules one at a time, as time permits, they can be quickly brought on-line.

To sum up, by carrying through some of these internal MIS systems you will be able to be more responsive to the needs and demands of the users. This will benefit the company as a whole, and your reputation in particular; making yourself indispensable is a good form of job security!

Finally, remember the main message of this paper: by automating MIS related tasks and thus saving time, you can 'be Proactive, not Reactive'.

5005

Adventures in Installing and Managing an International Network and/or How sure Can I Be With Netassure?

John R. Bedard
St. Jude Medical, Inc.
One Lillehei Plaza
St. Paul, MN 55117

COMPANY BACKGROUND

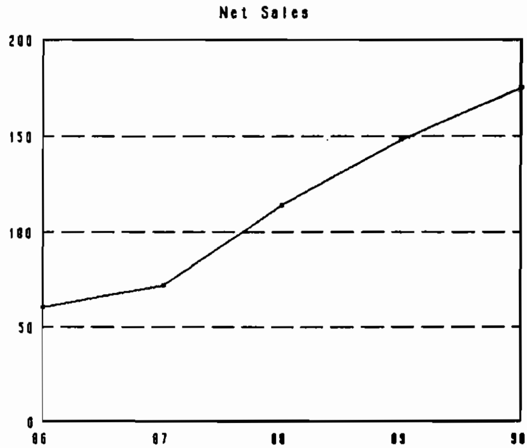
PRODUCTS: HIGH QUALITY PRODUCTS FOR CARDIOVASCULAR AND VASCULAR CARE, INCLUDING:

- PROSTHETIC HEART VALVES
- VASCULAR GRAFTS
- INTRA-AORTIC BALLOON PUMP SYSTEM
- CENTRIFUGAL BLOOD PUMP SYSTEM

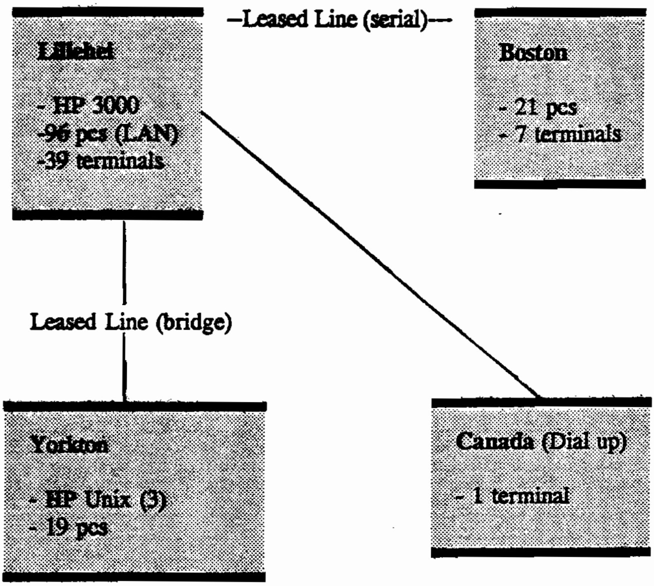
CUSTOMER SERVICE

- ORDERS RECEIVED BY 6:30 P.M. ARE DELIVERED BY 10:00 A.M. THE NEXT DAY
- NO BACK ORDERS
- ON CALL 24 HOURS A DAY, PROVISIONS FOR TWO HOUR EMERGENCY DELIVERY IN MOST LOCATIONS

COMPANY BACKGROUND (CONTINUED)

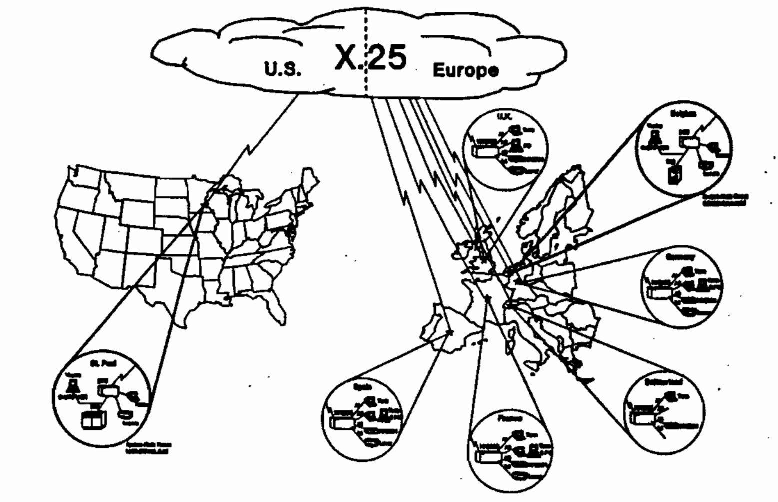


DOMESTIC DATA COMMUNICATIONS/LAN NETWORK



INTERNATIONAL DATA COMMUNICATIONS/LAN NETWORK

St. Jude Medical X.25 Network



Adventures in Installing and Managing an International Network
and/or How Sure Can I Be With Netassure?

WHY NETASSURE

- **NETASSURE IS SUPPOSED TO:**

  -Isolate Faults

  -Provide Multivendor Problem Management

- **STAFF LIMITATIONS**

  One person was responsible for:

  -Data Communications
  -LAN Support
  -Pc Support
  -HP 3000 System Manager
  -All Financial Systems
  -Human Resource System

- **MISSION CRITICAL DATA COMMUNICATION LINKS IN EUROPE**

  -Installation Date Critical
  -Customer Service Link Critical

- **CAREER CRITICAL DATA COMMUNICATIONS LINKS**

- **CONCERN THAT IT WOULD BE DIFFICULT TO ISOLATE LAN PROBLEMS**

INSTALLING EUROPEAN DATA COMMUNICATIONS

- **DEPEND ON GOVERNMENT PTT (POST OFFICE)**

  -Lost Paperwork

- **LEAD TIMES**

  -Plan on 3-4 months

  -Difficult to get firm installation dates

- **GOVERNMENT RESTRICTIONS**

  -Limitations in Spain

- **MODEM RESTRICTIONS**

  -Modems supported by HP not on PTT approved list

- **DIFFERENCE IN ATTITUDE TOWARD CUSTOMER SERVICE**

  -You expect that just because it is installed it should work?

- **INEXPERIENCE OF VALUE ADDED NETWORK**

- **HAVING HP INSTALL THE X.25 WAS THE SMARTEST THING I DID**

  -Which HP office prepares the NISP?

  -How do I get HP offices to talk to each other so I don't have to be in the middle?

TRYING TO GET A NETASSURE CONTRACT

- **WHO AT HP SELLS NETASSURE**

  -I think that they know now

- **HOW CAN I CONVINCE THEM TO SELL ME A NETASSURE CONTRACT**

  -It was not as easy as you would think

- **DO I HAVE TO PAY HP TO TAKE AN INVENTORY**

  -Why ?

- **COMPLICATIONS OF EUROPEAN OPERATIONS**

  -Do I need one or two Netassure contracts?

GETTING FULL VALUE FROM NETASSURE (U.S)

- **DEVELOPING APPROPRIATE COMPUTER ROOM PROCEDURES**

  -Initially Netassure doubled the work and elapsed time to resolve problems

  -Developed procedures for X.25 and leased line problems (copies available at conference)

- **OPPORTUNITY TO GET TO KNOW RESPONSE CENTER MANAGERS**

  -Initial Netassure calls were not handled very well at response center

  -We can tell if there is new staff at response center

- **MEETING WITH LOCAL HP STAFF TO REVIEW OUR EXPECTATIONS AND EXPERIENCES**

  -They needed to understand the performance required for Netassure to be of value to us

  -Led to revised procedures in our computer room which reduced our effort

  -Led to revised procedures at response center which help provided faster problem resolution

  -Decision to use agency letters with our other vendors

GETTING FULL VALUE FROM NETASSURE (U.S) (CONTINUED)

- **DEALING WITH PERCEPTION OF OPERATIONS STAFF THAT THEY COULD**

  RESOLVE PROBLEMS ON THEIR OWN FASTER THAN THEY COULD WITH NETASSURE

  -Insisting that procedures be followed

- **MEETING WITH VALUE ADDED VENDOR AND LOCAL HP STAFF TO REVIEW OUR EXPECTATIONS AND EXPERIENCES**

  -Value added vendor needed to understand our expectations

  -Led to revised procedures at value added vendor

- **MONITORING PERFORMANCE ON ALL DATA COMMUNICATION LINES**

GETTING FULL VALUE FROM NETASSURE (EUROPE)

- **OPPORTUNITY TO MEET SEVERAL LEVELS OF MANAGEMENT OF VALUE ADDED VENDOR TO EXPLAIN OUR EXPECTATIONS AND EXPERIENCES**

  -Had to teach the value added vendor's European staff to give us ticket numbers

  -Discussed situation up to and including the executive vice president

  -Our company is known by most of the value added network's European staff

- **DEVELOPING APPROPRIATE PROCEDURES TO REPORT PROBLEMS**

  -Needed to have both our European staff and HP European staff understand how we expected Netassure to work

  -Needed to establish a single point of contact between our European operation and the value added vendor

REPRESENTATIVE CHALLENGES WITH VALUE ADDED NETWORK IN EUROPE

- **COUNTRY SPECIFIC NETWORK RESPONSE CENTER**

  -Difficult to know which one to call

  -Different hours of coverage

- **HOLIDAY COVERAGE**

  -And they have holidays in Europe

- **NODE SITE NOT ADEQUATELY AIR-CONDITIONED (THREE DAY INTERRUPTION OF SERVICE)**

- **RELIANCE ON LOCAL PTT FOR LEASED LINE TO NODE**

  -Lacks sense of urgency

  -Difficulty of making changes in Spain

  -Technical competence ?

- **STAYING INFORMED ON PROGRESS UNTIL PROBLEM IS RESOLVED**

LESSONS I HAVE LEARNED

- MAKE EXPECTATIONS EXPLICIT WITH ALL VENDORS
- DEMAND THAT SERVICE MEET EXPECTATIONS
- INVOLVE VENDORS IN WRITING PROCEDURES
- HAVE CLEAR COMPUTER OPERATIONS PROCEDURES
- RECORD ALL PROBLEMS ON DATA COMMUNICATIONS NETWORK
- DO NOT HESITATE TO CONTACT HIGHER LEVELS OF MANAGEMENT IF EXPECTATIONS ARE NOT BEING MET
- NETASSURE CAN REDUCE, BUT NOT ELIMINATE, MULTIVENDOR PROBLEMS
- MINIMAL LAN PROBLEMS SINCE GOING ON NETASSURE

ATTACHMENT 1

REVISION DATE: 10/22/91 SAVED: TRC0624.OPSDOC.IS

FROM: TRACY CARTER

SUBJECT: CHECKOFF LIST TO TROUBLESHOOT THE LEASED LINES

PURPOSE: CHECKOFF LIST TO DOCUMENT LEASED LINE PROBLEMS

A. DATE OF PROBLEM:____________________ TIME:____________________

OPERATOR PERFORMING TROUBLESHOOTING:____________________

B. LEASED LINE CHECK LIST

LOCATION	MUX ID	AT LIGHT (YES/NO)	IF NO, OTHER LIGHTS	RESET (YES/NO)
Aries	M1	________	____________	______
Yorkton	M2	________	____________	______
		NORMAL - 56K BPS (yes/no)		
Lan 2	YKT	---------	-------------	-------

C. WAS IT NECESSARY TO CONTACT NETASSURE?____________________
DATE:____________________ TIME CALL REPORTED:____________

STAT MUX IS DOWN AT LOCATION:____________________

THE PHONE COURIER IS:____________________

THE STATUS LIGHTS ARE: (see above chart)____________________

D. NETASSURE CALL BACK NUMBER ASSIGNED:____________________

E. DOCUMENT HISTORY LOG IN REFERENCE TO THIS CALL:

Date Time contact name update ...

F. DATE NETASSURE CALL CLOSED:____________________ TIME:____________________

G. BRIEF EXPLAINATION ABOUT NETASSURE PROBLEM RESOLUTION:

____

____

H. Was the NETASSURE service satisfactory?____________________

If not, indicate problem areas:____________________

ATTACHMENT 2

REVISION DATE: 02/22/91 SAVED: TRC0118.OPSDOC.IS
TO: COMPUTER OPERATIONS
FROM: TRACY CARTER
SUBJECT: X.25 TROUBLESHOOTING CHECKLIST
==============================

PURPOSE: A CHECKOFF LIST TO GUIDE THE OPERATOR THROUGH THE TROUBLESHOOTING STEPS AND PROVIDE A HISTORY RECORD.

********* Follow procedure saved in TRC0117A.DOC.DP *********

A. DATE OF PROBLEM:_______________
TIME OF PROBLEM:_______________
OPERATOR INVOLVED:____________OTHER IS?__________________________

B. DID CONSOLE RECEIVE MESSAGE?__________ IF SO, ATTACH PSCREEN______

C. LIST ANY FLASHING LIGHTS ON MODEM:________________________________

D. LIST ANY LIGHTS OFFLINE ON MODEM:_________________________________

E. MODEM TURNED OFF & ON?_________________HOW MANY TIMES?___________

F. RECORD STATUS OF X.25 GLOBAL LEVEL: BOARD STATE=__________________

G. RECORD STATUS OF LEVEL 2: Level2 state=__________________________
Level1_________________________________

H. RECORD STATUS OF LEVEL 3: Level 3 state=_________________________
Trace status=__________________________

I. RECORD TIME NETASSURE CONTACTED___________________________________
NETASSURE TICKET NUMBER ASSIGNED:________________________________
RECORD THE TIME NETASSURE RETURNED CALL:_________________________
RECORD TIME DUTY MANAGER CONTACTED IF NECESSARY:_________________

J. RECORD TIME TYMNET CONTACTED:_____________________________________
TYMNET TICKET NUMBER ASSIGNED:__________________________________

K. BRUSSELS CONTACTED:____

L. TIME THE X.25 CONNECTION BECAME AVAILABLE:________________________

M. PROBLEM RESOLUTION DETAILS:_______________________________________
____
____
____

N. DATE TYMNET CALL CLOSED:____

O. DATE NETASSURE CALL CLOSED:_______________________________________

P. WAS TROUBLESHOOTING SUPPORT SATISFACTORY?_________________________

Paper Number: 5006

Presentation Title: Applying Mainframe Security and Audit on the HP 3000

Lee Courtney

Monterey Software Group

Cupertino California

408-253-1778

"The minicomputer is dead! Long live the minicomputer!" is a cry often heard today. With personal computers offering superior price performance and ease-of-use, and mainframes firmly entrenched in many MIS shops, the minicomputer sometimes looks like it is headed in the same direction as the punched card. But despite predictions of the imminent demise of minicomputers, many vendors are continuing to see respectable growth for both proprietary and open systems. For example, last year Hewlett-Packard reported over ten percent growth for its HP 3000 product line, with the installed base surpassing 50,000 systems.

Minicomputers exhibit features found in both PC and mainframe environments. From the PC environment minis share ease-of-use features and approach the PC level of price-performance. At the high end minis share mainframe traits such as the ability to function as an organization's information resource processing and storage utility. A good example of the minicomputer environment today is the HP 3000 and its MPE operating system. While technical specifics may not be applicable, many of the general concepts concerning the HP 3000 are germane to other platforms, be they a DEC VAX or an IBM AS/400. Thus in this paper it is assumed that the HP 3000 can effectively represent the minicomputer spectrum and it will be used to illustrate general security characteristics of minicomputers.

While minicomputers approach mainframes in terms of system throughput, the number of users supported, and storage capacities, they do not commonly possess the same sophisticated administrative and security facilities. What are the security and audit problems facing minicomputer sites? To answer this question it is helpful to look at the evolution of the HP 3000 product line, and examine the security capabilities of the MPE operating system.

The HP 3000 was introduced in the early 1970s as the first interactive multi-user business minicomputer offering COBOL, a bundled database management system called IMAGE, and a easy to use terminal forms package called V/PLUS. At that time most HP 3000s were stand-alone systems running a single application for a homogenous user community. The security strategy was one of controlling access based on the concept of an "insider" verses an "outsider". An insider was a person within the organization with permission to use the system, while an outsider did not have permission. The only means of enforcing this control was logon passwords.

The next phase of HP 3000 maturation began in the early 1980s with the introduction of the first system capable of supporting over 200 on-line active users. By this time networking was becoming standard and included both peer-to-peer as well as HP 3000-to-mainframe connections. Users now had the CPU power and features to greatly expand where and how the HP 3000 was used in their organization. It was not unusual to find Fortune 500 companies using the HP 3000 in a distributed processing environment with extensive networking, while medium and small businesses used the HP 3000 as their central information utility. Applications running on the HP 3000 began to include electronic mail, accounting systems, database management applications, document preparation, and program development. Thus this period was characterized by (1) both networked and stand-alone HP 3000s, and (2) a variety of usually centralized applications.

Today the HP 3000 product line has evolved from that first stand-alone minicomputer to a family of systems capable of supporting up to 1000 users. In the late 1980s HP introduced the first commercial system based on a Reduced Instruction Set Computing (RISC) architecture. Almost all HP 3000 systems are integrated into some type of network (including networks of PCs as well as peers and mainframes) and run a diverse set of applications. For many businesses they perform the same role as mainframes may have performed previously, but with improved price-performance and ease-of-use.

Concurrent increases in the power of the HP 3000 and in the integration of computer processing into daily business operations has made computer processed data the most valuable asset of an organization. However, the security functions offered by the MPE operating system have not grown commensurately with the value of the information stored on those systems. In 1992 the primary control used to secure HP 3000 systems is still a password at logon time.

How can security, access control, and audit concerns be addressed in a manner that will preserve HP 3000 ease of use, compatibility, and price-performance? Mainframe systems are characterized by a history of well developed commercial security standards and awareness, along with IBM and third-party security products. Thus one can examine solutions successfully used in the IBM MVS and VM environments, and apply those concepts which would be beneficial.

Which standards in the mainframe environment might be utilized on the HP 3000? In general these include individual accountability, auditability, and separation of duties. Individual accountability provides the capability to tie an event on the system, such as a logon or file open, to a specific individual. Closely tied to individual accountability is the concept of auditability. Auditability insures that the system collects the information necessary to accurately determine who is using the system and what they are doing. For example, an auditable system would record who unsuccessfully attempted to open the payroll database. Separation of duties can be used to implement checks and balances on the system and thus acts as a deterrent to abuse.

In addition to the above general concepts, some implementation specific characteristics are also essential to insuring a mainframe level of security on the HP 3000. First is the concept that security and audit controls should be integrated at the operating system level rather than at the application level. Second, security and audit controls must apply to all classes of users on the system. Increased security should not require changes to applications and should be compatible with operating system and network interfaces. In other words increased security should not be disruptive to users, applications, the network, or production work.

How can desirable mainframe standards be applied on the HP 3000 today, and what additional capabilities are required to secure an HP 3000? Mainframe security falls into three areas: system access control, file access control, and audit. System access control is concerned with the identification and authentication (I&A) of users when they first establish contact with the system. This includes both interactive access through terminals and batch access through jobs, along with access through network protocols. File and database access controls are concerned with controlling access to both programs and data files by authorized users. Detection of unauthorized access attempts, and verification of authorized access is controlled by the audit function.

"Of Course We're Secure, All My Users Have Passwords!" - System Access Controls

The first security mechanism users encounter regulate access to the system through interactive sessions or batch jobs. Like mainframes, the primary system access control on the HP 3000 is the password. However, while MPE provides passwords at the user, account, and group levels, it does not provide any password composition or administration controls.

To extend the capabilities of the rudimentary password protection provided by MPE, some security implementations use a Logon User Defined Command (Logon-UDC) to implement I&A controls. While this method may have been adequate in the past, its use today is a severe control deficiency. The problem with Logon-UDCs is that authentication takes place after a user has logged on the system, not before. This would not be a problem except that the Logon-UDC mechanism may be disabled by accidental or malicious action. If this happens, there are no system level access controls in place at all! Logon-UDCs are also incompatible with network file transfers since they prompt for passwords differently than MPE. While not security related another concern is the impact Logon-UDCs have on logon performance. On systems where Logon-UDCs are utilized the I&A sequence can take up to 20 minutes to complete.

In the mainframe environment one finds the same identification and authentication checks, but more comprehensive and efficient password and administrative mechanisms. These include extensive password controls to prevent unauthorized access as well as features to control access by legitimate users. Password controls in mainframe environments greatly increase the usefulness of passwords. These controls integrate I&A, password aging, reuse and composition controls at the operating system level and enable user password administration.

Integration of password checks at the operating system level is one of the most important aspects of mainframe environments. Users encounter the password verification process before gaining access to the system. In addition, when implemented at the system level, rather than at the application level through Logon-UDC schemes, the performance and resource impact of the I&A process is greatly reduced.

In order to increase the effectiveness of a password system, many mainframes implement extensive password controls. Password aging and reuse controls force users to periodically make use of 'fresh' passwords to reduce password exposure. Along with password aging, mainframe security products also allow a security administrator to setup standards for password composition. The ability to setup such standards should be an integral part of any HP 3000 security program.

While password controls prevent access by unauthorized users, in many mainframe environments we find mechanisms which are also intended to further enhance access controls for authorized users. These controls include both user and device logon restrictions. User controls define the how and when of access to the system. Device controls are used to specify where a user is allowed access to the system. Typically these controls are realized using a general access rule language. Such controls can be an effective deterrent against insider abuse by restricting a user to only permissible times, days, and terminals.

System access can also be gained through batch jobs. While batch environments are different between the HP 3000 and mainframes, the use of some general principals can increase job security. The first is to use facilities that eliminate the need for embedded passwords in batch jobs. Embedded passwords greatly increase the probability of password exposure and make secure job sharing difficult, if not impossible. Second, when batch jobs are shared between users, users should make use of facilities which completely eliminate password sharing and prompting. Use of access rules not only provides a secure job submission and sharing facilities, but also increases the productivity and flexibility of batch job streams.

The purpose of system level access controls on mainframe systems is to prevent unauthorized access and to control access by authorized individuals. The goal for HP 3000 system managers and security administrators should be the same. System management should make sure that they are taking advantage of technology which provides baseline password composition and administration controls, and is tightly integrated and compatible with the operating system and networking protocols.

"My Users Don't Know Enough To Do Any Harm" - Program and Data File Access Controls

One of greatest security risks on the HP 3000 today is a deficiency of effective and easy to use file access controls. These access controls are the mechanisms governing which files and databases authorized users may access once admitted to the system. Given that the greatest potential for computer abuse is from users having permission to access a system, file and database access controls are a critical requirement. Studies consistently show that the vast majority of computer crime and security related events are perpetrated by users who are authorized to access the system. Given this fact, implementation and use of controls which enforce individual accountability should be mandatory.

Until recently the absence of effective file level access control facilities, especially at the operating system level, has been a major security deficiency on the HP 3000. In the mainframe environment we find several products which define baseline standards in file access control. The two most widely used being RACF(from IBM Corporation) and ACF2(from Computer Associates). These products provide a mechanism called access rules which can be used to clearly define file access and sharing in multi-user environments. Access rules define the how, when, where, and who of file and database access.

On the HP 3000 if file access controls are implemented at all, it is common that they are embedded within an application. However, when implemented at the application level, controls are limited to those programs that have been explicitly modified. Moreover, putting application programmers in the security implementation business violates well accepted standards concerning separation of duties. And even with these difficult implementations, the resulting application controls give only the appearance, rather than the substance, of security. Using system level file access controls removes these problems by providing a comprehensive set of controls which apply to all users, programs, and data without the need for application changes. The controls even apply to access by the operating system itself, thereby controlling all access to sensitive data and applications.

At the system level MPE provides several mechanisms to control file access, but these mechanisms are difficult to use and maintain, contradict one another (there are 3 separate ways to share a file under MPE), or cannot be used to enforce individual accountability.

The most widely used sharing mechanisms are the RELEASE command and the access matrix. But neither of these mechanisms is capable of enforcing individual accountability since they cannot allow a file to be shared based on a logon-id. The RELEASE command allows all users unrestricted access to the file until the owner of the file explicitly invokes the SECURE command to retract such access. The access matrix allows sharing at the group, account, or system level, but not by individual logon-id. MPE Access Control Descriptors(ACDs) allow flat files and devices to be shared based on an individual user's logon-id, but cannot be used to control IMAGE databases and are limited to the file level rather than the file, group, and account levels.

Third-party access rules can be used to obtain the same degree of control as available on mainframe systems. Access rules allow flat file, IMAGE database, device, or other objects to be shared in a secure and auditable manner. Implemented at the operating system level, access rules do not require any application changes. As on mainframe systems sharing can be based on a user's logon-id, access type, source of access, time, or date.

In addition to providing increased security, file access controls permit greater flexibility in the type of applications and users supported by the system. Effective controls permit the introduction of applications which may have previously presented an unacceptable risk. Utilization of mainframe standards and techniques allows introduction or expanded use of applications such as electronic mail, customer service programs, bulletin boards, and Electronic Data Interchange (EDI). Thus the use of mainframe access controls not only increases system security, but also system flexibility and productivity.

"What I Don't Know Can't Hurt Me" - Audit Data Collection, Reduction, and Analysis

Perhaps the weakest area of control on the HP 3000 is verification of authorized system access and detection of unauthorized access attempts. This weakness is due to a lack of tools and audit data.

While application auditing is widely practiced on the HP 3000, some administrators may not be aware of the need for auditing at the system level. The primary system audit function performed on the HP 3000 is the configuration audit. A configuration audit specifies which system settings may lead to security exposures, but does not report actual system activity. On the other hand, an activity audit allows review of events which occurred on the system. Configuration audits report what might happen, while activity reports describe what has occurred.

Although activity reporting may be performed on the HP 3000 using system log files, they have several shortcomings which limit their usefulness. Often the data collected by system log files is not comprehensive. The MPE utility used to display log files allows selection of audit events by logon-id, but does not provide reporting by program or data file name. This approach is not efficient since it provides user rather than file based reporting. While reporting by user is important, the ability to examine program and data access is even more critical. Of course the ideal system will contain both functions.

The mainframe environment offers several important characteristics which are applicable on the HP 3000. When we look at mainframe environments we find not only configuration audits, but also tools and data to monitor system activity from a security and control perspective. Most significant is the integration of audit data collection at the operating system level. This insures that all security related events are captured for all classes of users. As a result system management and application development staff are subject to audit review, not just end-users.

One of the shortfalls of having comprehensive data collection, is that reports often contain more data than can be easily comprehended. To increase audit productivity, data reduction and analysis tools must be used. Audit data reduction and analysis tools extract only those audit events fitting user defined selection criteria. These machine based filters greatly reduce the audit workload, increase productivity, and raise the confidence level in audit results.

The audit function is a critical element of any security program. It is the methodology by which management confirms appropriate system use. In the last year tools have become available which provide mainframe type system audit functions with minicomputer ease of use. These tools allow a user to examine the state of the system and look at system activity. Using these tools to implement mainframe audit standards and techniques, users of the HP 3000 can insure the accuracy, integrity, and availability of system and information resources.

Conclusion

What should we look for in a secure minicomputer environment today? Given the roles minicomputers are playing in many organizations, it is important to use those techniques and methodologies which have proven successful and are applicable. Solutions which provide standards concerning individual accountability, auditability, and allow separation of duties should be implemented. To be effective, security tools must be integrated at the system rather than the application level, must provide comprehensive controls which apply to all user populations, must protect against inside abuse as well as outside attacks, and must furnish a complete audit trail of system use and configuration. Given the current state of HP 3000 security, tools which implement mainframe standards and features represent a quantum leap in information asset protection. With these elements in place, one is well on the way to assuring a secure and reliable minicomputer environment which can withstand the tests of contemporary security threats.

Lee Courtney is President of the Monterey Software Group in Cupertino California. At Monterey Software he is involved in the design, development, and marketing of high performance system management tools for Hewlett Packard computer systems. Prior to joining Monterey Software, Mr. Courtney was at Hewlett-Packard where he was the Product Manager for the MPE V/E and MPE XL operating systems, and worked as an R&D engineer on the design and development of operating systems for the HP 1000, Vision, and Spectrum computer families.

downsizing

one side will make you
grow taller,
and the other side will
make you grow shorter

robert a. karlin
karlins' korner
1628 van noord ave.
n. hollywood ca.
91605

(818) 982-9331

One Side Will Make You Grow Taller, And The Other Side Will Make You Grow Shorter

5007-1

The Caterpillar took the hookah out of its mouth, and yawned once or twice and shook itself. Then it got down off the mushroom, and crawled away into the grass, merely remarking, as it went, "One side will make you grow taller, and the other side will make you grow shorter."

"One side of *what?* The other side of *what?* ", thought Alice to herself.

Alice in Wonderland
Lewis Carroll

INTRODUCTION

Wouldn't it be nice if we could find a small bottle filled with blue liquid that says "Drink Me"? We could give it to the MIS department, and watch the MIS budget shrink to a manageable size.

Sadly, that option is not open to us. So, in an effort to curb an MIS department that in many cases is one fifth to one third the entire corporate budget, firms are turning to PC networks to replace the centralized mainframe, usually without contemplating the actual costs or consequences.

This paper will examine the capabilities of PCs and mainframe computers, discussing the strengths and weaknesses of each. We will also discuss the current state of distributed processing as a possible solution to MIS budget woes.

During the following discussion I will be referring to "Mainframes". I am using this term to refer to all classes of multiuser systems, from minicomputer to supercomputer. The line between differing classes of multiuser systems has become so blurred that any attempt to separate them is beyond the scope of this discussion.

PC vs MAINFRAME
Hardware

Comparing PCs to Mainframes is somewhat difficult, since many of the metrics used to compare differing hardware have not been applied to PCs. Does a 40 megahertz PC compare to a 40 MIP (Million of Instructions per Second) mainframe? No, not really. The instruction rate of the PC depends on the width of the bus used to read the instructions into the processing chip. If the instruction is 32 bits long, and the memory bus is 16 bits wide, it will take two fetch cycles to bring the instruction to where it can be decoded. Also, each instruction may take anywhere from 2 to 500 clock cycles to execute.

It is easier to compare bus speeds and peripheral data access time. However, there are still pitfalls in comparing mainframes with PCs. Many PC peripheral cards will interrupt the PC microprocessor for each word transferred, whereas most mainframes depend exclusively on Direct Memory Access (DMA) transfers that transfer data directly to computer memory without interrupting the central processor (CPU).

With this in mind, we will attempt some comparisons. The source for most of the information listed here is the 1992 Data Sources. It should be noted that by the time this paper has been printed there will be new machines and different and probably lower prices.

The largest IBM computer available in my sources is the IBM ES/9000 model 900. This computer has a maximum memory size of 9.2 gigabytes, and will run at 220 MIPS. List price is about 23 million dollars. The maximum number of users is probably about 4000. The largest HP machine listed is the HP9000 series 800, model 870s/200. This machine has a maximum user count of 800, runs at 100 MIPS and will support 768 MB of RAM. List price $689,000. The fastest PC that is now available uses the INTEL 30486 CPU chip running at 40 megahertz. List price would probably be $10,000 for a fully loaded system. Conservative calculation puts the 486 at between 1 and 5 MIPS, supporting from 1 to 10 users (depending on the operating system). Our calculations, if we just use MIPS, would be $100,000 per MIP for an IBM mainframe, about $7000 per MIP for the Hewlett Packard, and $2000 to $10,000 per MIP for PC.

What does this tell us? Virtually nothing. Not even the hardware vendors can agree on what a MIP is, or really means. Each hardware vendor has developed a different standard for measuring throughput that is designed to accentuate the advantages of its particular machine. A standard benchmark is difficult to develop, since it is probable that the mix of processes will not match the use that the machinery is destined for. The only way to see how a specific application will perform on differing hardware platforms is to run it on each platform and compare the results. Since designing an application on a hardware platform necessitates purchasing or leasing that platform and then expending the development effort, this advice is like telling a prospective car buyer to buy one of everything, drive them all for 5 years, and then choose the car he likes.

It seems obvious that other factors need to be used to select a platform other than hardware cost verses performance. Disk speed and capacity can be important in deciding which system to choose. Large IBM systems generally use the IBM 3380 or 3390 subsystems. These disk units, using large 14 inch platters, and having maximum capacity of 19,000 megabytes for a subsystem, have an access time of 12 to 13 milliseconds, and a maximum transfer rate of 4.5 megabytes per second. Hewlett Packard offers the series 6000 drives with 27 millisecond access time, and 1.2 megabytes per second transfer rate. The current PCs can support internal drives with over 500 megabytes with access times of 20 milliseconds, and smaller drives in the 50 to 200 megabyte range with access times of 14 to 15 milliseconds. The average transfer rates for these units is anywhere from 5 to 15 megabytes per second.

Again, these figures are misleading. Cache, both in the hard drive and in the central processor, can speed disk access enormously for certain applications. The mix between random and sequential retrieval is also extremely important in determining disk performance. The amount of time that the central processor needs to be involved in each data transfer is also of paramount importance in discussing disk access. Channel and controller contention (the channel is the microprocessor in mid to large range computers that controls the data path between the controller and main memory when the CPU is not involved) must also be factored into the equation.

Even factoring in the appropriate disk access and transfer rates to the mainframe speed does not tell us how a particular application will function. Operating systems provide overhead that must be considered, and that overhead will be different depending on the type of processing that the application or mix of applications require. Evaluating hardware in determining the future course of a particular data center should not be a primary factor, though it can provide indications that can be used when the final projected application mix is known. In fact, the decision as to which hardware platform is appropriate should be made last, after all other considerations has been taken into account.

PC vs MAINFRAME
Operating System Environment

Operating systems provide the fundamental backbone for all data processing applications. The strengths and weaknesses of the operating systems provide one of the more important metrics on which data system to purchase. We will examine each operating system separately, and then compare them.

Most large IBM facilities run under IBM's MVS operating system. This operating system provides the facility for running batch jobs. In order to provide for online access, two other products are necessary, TSO and CICS. TSO is a teleprocessing monitor used mostly as a development environment and production support tool. Another product, ISPF, provides a screen and menu oriented overlay to TSO, including an integrated editor and file maintenance program. CICS is used as an application teleprocessing monitor. The integrated screen handler is cumbersome, and the coding style required to prevent CICS from bogging down is downright obscure. On the other hand, CICS supports an enormous number of users compared to TSO. The batch job control language is initially somewhat difficult to learn. One of the major advantages of MVS is that there are many pieces of third party software designed to overcome deficiencies in the operating system and online environments. One of MVS's major drawbacks is the number of systems programmers necessary to maintain the system.

The HP business operating system MPE, on the other hand, provides an integrated batch and online environment. Unlike IBM, there is no difference in the control language between batch jobs and online sessions, and programs written for batch may be run online with no changes. The HP full screen handler is also cumbersome to use, but forms design is slightly easier than on an IBM mainframe. MPE is also much easier to maintain. Many shops can exist with minimal support, either a part time consultant or an employee who spends part time on system management. Also, unlike IBM, HP provides an integrated data base, either IMAGE, HP's network data base, or HP-SQL, an industry standard data base. HP also seems to be moving MPE away from one of its major drawbacks, the inability to handle industry standard but non HP standard peripherals. The major drawback to HP's MPE is the lack of sophisticated data archival tools, such as tape management, or generation data sets. Also, HP's reluctance to embrace ASCII standard labeled tapes has made third party integration of these products unpredictable.

The current fad in operating systems is UNIX. UNIX is touted as providing a standard operating environment across hardware manufacturer's boundaries. Versions of UNIX run on large IBM and Hewlett Packard mainframes, as well as on PC's. In many ways, however, this standardization is somewhat fictitious. At the hardware level, there are many differences which can affect software performance. In designing software for a particular environment, care is usually taken to optimize the software for the environment in which it is running. Transferring that software from one hardware platform to another usually means a degradation in performance, no matter how 'standard' the two platforms are.

UNIX provides many more tools to aid in software development. UNIX supports XWINDOWS, a Graphical User Interface much like Windows under DOS. UNIX is also extremely flexible, allowing programmers to take advantage of many features at an extremely intimate level within the operating system. Though most programmers conversant with UNIX that I know love developing under it, UNIX's flexibility provides many openings for security breeches and catastrophic software failures. The UNIX operating environment can also be complicated for new users, due to the multiplicity of commands, and the lack of any concrete standard in how these commands work.

Most PCs today run Microsoft DOS operating system. DOS does not provide for any real form of Screen Handler, though this gap is being taken up by Microsoft Windows. Windows provides a GUI

(Graphical User Interface) that is much more flexible than most mainframe users interfaces, allowing more sophisticated displays than is possible on those interfaces. Windows also provides a slightly more sophisticated memory manager, though in most cases, protected memory is not used, due to its adverse impact on machine speed. The biggest advantage of DOS is the lack of contention from any other user, and its biggest disadvantage is its inability to share data easily, and the instability of the operating environment, due to the lack of standardization amongst software packages. In order to share data, a separate software package and usually a separate file server machine, must be integrated into the environment. The most common network software package is Novelle. Novelle provides file server capability and file security across the network. Its main drawback is the difficulty in properly configuring a Novelle network, and the expense of the File Server PC.

Another operating system for PC's is OS/2, also a product of Microsoft and distributed by IBM corporation for its PS2 line of computers. OS/2 addresses many of the drawbacks of DOS, including an integrated LAN manager and screen handler, but OS/2 needs a fairly large, fairly fast PC in order to run. IBM has also made it fairly expensive to develop on, due to the price of the OS/2 developers kit. Unless there is a specific package that you need that only runs under OS/2, I would not recommend choosing it.

Looking at the choices listed above, the decision as to which hardware and software platform becomes a little clearer. The large IBM boxes, along with the MVS operating system, are designed to move and store large masses of data. Any system that is planned that needs efficient archival storage of data is also a target for an IBM style mainframe. Systems that have need of efficient online access to data, as well as batch access to that data can focus on a midrange mainframe, such as the HP3000 under MPE. Standalone PCs are extremely useful when there is little need for access to large data bases. Again, unless you are developing software, or wish to buy a package that only runs under UNIX, I would not recommend purchasing it.

PC vs Mainframe
Application Software

Most software that is used in today's businesses are packages, that is, they are generic forms of whatever software is needed, such as General Ledger or Accounts Payable software. Most businesses can use any number of packages to run their day to day operations, however, there is always something that is unique to their particular business niche. Many times some portion of a business's production software must be custom written to provide information that is not found in a generic package. Some packages provide the facility for the business to modify their software to accommodate these requirements. Other times, the software must be written from scratch. When a package provides enough flexibility to forgo expensive development, a business will consider moving to the particular hardware platform that the software was written for.

In choosing package software, there are a number of considerations that should be examined prior to purchase. The primary consideration, after determining that the package meets the necessary business requirements, is the recoverability of data in the event of a hardware or software malfunction. Is the data from the package stored in such a way that it can be easily backed up? Can the backup be used to trace software or data entry problems if necessary? Is there enough of an audit trail to determine if an entry error has occurred? If the data is lost, what would it take to recover? Are there transaction reports that can be used to reenter the data? Is any form of transaction detail kept to allow an auditor to prove the final reports?

A second consideration is the accessibility to the data kept by the package. Does the package allow data to be extracted and passed to other packages? Does the package allow data to be extracted from other systems and easily entered into this one? Can selected data be extracted for generic 4GL reporting applications?

Another consideration must be the quality of support provided by the software distributors. Will the distributor provide the support necessary to correct program errors in a timely fashion? Is there user support if there is question about how the software works? Does the distributor provide any training for the software? Does the distributor provide upgrades to new product versions?

And, finally, will the package grow with your business application, or, in purchasing this package are you locking yourself to software that will constrict the ability of your organization to provide service to your customers and users.

PCs and Mainframes
Distributed Processing

Within the next few years, a revolution in software will occur that will change the way applications are developed, and the way computers will look to the end user. As more users become familiar with PC software, especially under Microsoft Windows, mainframe software will be much to cumbersome to use. On the other hand, PC software will be inadequate to handle applications that must cope with large masses of archival data, or provide instant access to data entered at distant locations. For these applications, distributed client server applications will be the answer.

Most current applications that function across multiple PC hardware platforms and PC/Mainframe links are file server applications. This means that the local PC is executing a program that will read data from across some form of communications link, with all the processing occurring at the local PC. Though this allows access of non-local data, transferring data that may not be needed to the PC ties up the communications network, limiting both the number of users that can effectively use the network, and the amount of data that can be transferred.

The alternative to this would be to do a certain amount of processing at the file site, prior to transmission. Selections that involve large files and a small amount of selected data could be accomplished more efficiently. The tools to do this type of processing do not yet exist on most platforms, and those that do are not necessarily compatible with the tools from other platforms. Within the next few years, most hardware manufacturers will be adopting standards that will allow communications between programs on different platforms easily. As these tools become more common, applications will take advantage of the GUI interfaces on the PCs to access the large data stores of the mainframe. The PCs will process local data, edit user input and format the user screen. The mainframes will handle data collection, data base update, and archival storage. Amongst other benefits, this provides a secure and auditable environment for corporate data, without cluttering up the corporate data stores with data that is unneeded at the corporate level.

So What?

Have we learned anything from this walk through current data processing technology?

Well, first, if you are currently on a hardware platform, it is probably cheaper and easier to stay there than to switch, as long as your hardware manufacturer has not declared Chapter 11. The only impetus to switch would be finding a software package that meets a specific corporate need and runs on no other hardware. If this happens, give serious consideration to existing on multiple hardware platforms. Most current platforms can be linked into the same LAN (Local Area Network) without much difficulty. This would allow you to decide which platform most meets your requirements for the future, without a serious conversion effort.

If you must select a new platform, choose the software that you need first, and then choose the platform that runs it. Select a system that will allow your shop to grow. Find other data processing shops that run the mix of hardware and software that you expect to run, and ask them these questions:

1) How many people does it take to keep the system running? This includes operators, managers, programmers, etc.

2) What kind of environment is needed for the hardware?

3) What kind of hardware service is needed on a regular basis? Is the hardware support adequate?

4) Has the system ever crashed hard? If so, how long did it take to recover and why?

5) How easy is it to install software? What does it take to reconfigure your hardware and software for new devices and users?

6) Have they ever upgraded their system? How long did it take and what was required?

Also, have the software examined by both data processing professionals, and accounting and audit professionals to determine if the software was written with sufficient backup and audit features. They should also call any software vendors and report on the quality of support that they receive. If possible, hire a consultant that has no future interest in the project.

Finally, if possible, run the application with your data on the target machine. Note the response time and the usability in a pseudo production environment.

Good luck, and stay away from those mushrooms. Some are poisonous.

Paper # 5501

Performance Management Made Easy

Wayne Morris
Hewlett Packard Company
Performance Technology Center
8050 Foothills Boulevard
Roseville, CA 95678

Abstract

Performance management has traditionally been reactive in nature, and required a high level of expertise to identify and resolve performance problems. This doesn't have to be the case. Performance management can be easier and less time consuming.

This paper outlines an approach to performance management based on management by exception techniques which identify and highlight actual and potential performance problems. Once a problem is identified, performance rules are evaluated automatically to assist in characterizing and resolving the problem. This approach enables more rapid resolution of performance problems before user productivity is affected, and allows performance management expertise to be used more proactively to support business goals.

Introduction

Before we can examine how to make performance management easy, we should first define what it is, and why we would want to do it at all. The objective of performance management within an organization is to accomplish the following:

- ensure user productivity is not negatively impacted by the service levels (response times, job turnaround times) provided by the computer facilities;
- optimize the return on investment in the computer facilities;
- plan for support of future business goals and strategies by the computer facilities.

Computer facilities refers to the computer network, systems and applications that together automate one or more aspects of the organization's business. Performance management is the science of examining the way in which the business uses the computer facilities, characterizing the inherent performance capabilities of these facilities, optimizing the usage and availability of the facilities, and predicting future usage and capacity requirements in order to meet the objectives outlined above.

Traditional Performance Management

Traditional performance management is primarily reactive in nature. Users experiences slow or erratic system response, or perhaps their batch jobs haven't completed within an acceptable time, and they notify the help desk or system manager of the problem. Once a performance problem is identified, resolving the problem requires a great deal of expertise. Performance measurements have to be obtained to quantify and characterize the problem.

Generally, these metrics are very detailed, and it can be part science and part art to determine which metrics are important and to follow a logical approach to isolating and characterizing the actual problem cause. This is time consuming at best, and becomes very inefficient when distributed systems and applications are involved, as we now have to examine detailed information for each of the systems supporting the application.

Waiting for a problem to be identified by the user community implies that user productivity is already impacted before we begin to diagnose and resolve the problem, and consequently the profitability of the business can also be negatively affected. Because of this, the problem resolution will probably be undertaken in high-priority crisis mode, increasing the pressure on the facility manager and the performance analyst.

This leaves little time for the more proactive aspects of performance management such as examining workload patterns, scheduling jobs to take advantage of periods of low usage, and balancing utilization levels within classes of resources. Operating in a reactive mode also means that capacity planning is rarely undertaken, and hence as the business changes, it is probable that the computing facilities will not be able to support new requirements such as additional applications, users and workloads.

When compared to the stated objectives for performance management, this traditional approach is not really successful. User productivity is affected since performance problems are first recognized by the users themselves. We can't optimize resource usage since we aren't proactively examining trends in workload patterns and there is little attempt to balance workloads and resource usage. Finally there is very little time to plan for future requirements because of the reactive nature of performance management.

New Concepts for Performance Management

Two key concepts which can be applied to performance management are management-by-exception and rules-based automated analysis. Each of these concepts can be used to achieve the objectives of performance management more effectively and efficiently, lowering the overhead and reducing the expertise requirements.

Instead of waiting for users to be affected by performance problems, or actively monitoring all of the resources and applications within the computing facilities, we can focus on areas requiring attention by looking for deviations from a normal or desired state. This technique, referred to as management-by-exception, is not a new concept and has been applied to many management areas. When implemented for computing facilities management, it generally involves some form of alarm notification based on pre-determined thresholds being exceeded.

The second concept is to take the rules an expert would apply in resolving performance problems, and automate them to provide first level performance diagnosis, reducing the people overhead of performance management.

Both of these concepts are readily applicable to performance management and together increase the effectiveness and efficiency of the computer facilities' management staff. Although related, the two concepts achieve different purposes. Management-by-exception identifies performance problems within the computing facilities, rules based analysis assists in determining and correcting the cause of the problem.

We'll now examine how to implement these concepts. The first step in determining if an exception condition exists is to define the normal or desired state.

Defining the Desired State

To define the desired state, we need to examine the goals and objectives of the business, and in particular, to take the user's perspective of performance. Users will generally know when the computing facilities impact their productivity, but may have some difficulty in clearly articulating their requirements. The process of interviewing users to determine and document the required service levels is part of the process of establishing Service Level Agreements (SLAs). More information on service level agreements can be found in [McBRIDE] and [MILLER].

An SLA documents the user requirements and services to be provided in terms of application and system availability, response times, job throughputs, quality measures, anticipated workload volumes and user-projected growth in applications and workloads. This provides for a clear understanding of the services to be provided by the computer facility, and allows the facility managers to determine a set of measures and goals which defines the desired or normal state. These service level objectives are typically defined as:

- response times will be less than or equal to A seconds B% of the time
- batch jobs will complete within C minutes from submittal D% of the time
- the system will be available E% of the time

Each of these measures will normally be defined on a per-application basis and may also specify the measured shift, for example the response time objective might be stated for the prime shift rather than across the entire day. The service levels required for a financial dealer application may be much higher than those required for an inventory control application, and so the objectives reflect the relative importance of each application to the organization.

Another consideration may be optimizing the utilization of resources within the computing facilities. Under certain conditions, this goal can be contradictory to maintaining user productivity, and hence this may become a secondary objective. We may define the secondary aspect of our desired state as:

- resource utilization will not fall below A% more than B% of the time

This measure would be required for each class of resource and would be measured globally rather than on an individual application basis.

Now that we have defined our desired state, the next step is to continuously monitor the actual performance of the computing facilities to determine when there are significant deviations from normal, otherwise known as exception conditions.

Measuring Performance

Monitoring the performance of the computing facilities requires access to performance metrics captured from operating systems, intelligent network devices, and potentially from applications.

In the case of systems, this requires the operating system to be instrumented to provide performance information about processes, applications (groups of processes), and global resource utilizations. A number of proprietary operating systems have been instrumented and performance monitors are available which collect and log these performance metrics. In the UNIX* arena, standard utilities such as /dev/kmem are generally supported which also provide performance metrics. However, these metrics may not be totally validated and may not be as comprehensive as those provided on proprietary systems.

The number of metrics and their level of accuracy and validation will vary from platform to platform, so managing the performance of multiple heterogeneous systems is more difficult than managing a homogeneous environment. [MORRIS]

The performance of many intelligent network devices can also be monitored. These devices typically hold their metrics in a Management Information Base (MIB), and access is provided through the Simple Network Management Protocol (SNMP). [ROSE]

The performance collection should provide the required detail of information, and the collection and logging process itself shouldn't impose too high an overhead on the system. It is also important that the performance metrics allow service levels to be evaluated from a user perspective. Users are typically concerned with an application or particular transaction within an application. Therefore, the collection mechanism should support the grouping of process information into defined applications that correspond to user terminology such as the payroll application or order processing application.

Similarly a mechanism by which a user transaction can be defined, measured and reported may also be important in determining if user service levels are met. This mechanism could either be provided by allowing application transactions to be marked by an operating system intrinsic which can then be identified and logged as part of the normal collection process, or by allowing entries to be written into the performance logfile directly by an application.

As the performance measures are collected, we need to be continuously comparing against our stated service level objectives and resource utilization objectives to determine if an exception condition exists.

Determining if an Exception Condition Exists

The simplest form of determining when an exception condition exists would be to compare our measured performance in terms of response times and job throughputs against our service level objectives, and when the objectives were exceeded to trigger an alarm. This sounds too easy to be true, and the situation is generally more complex than this. This is particularly true of UNIX-based systems, where the definition of response time is very inconsistent.

Users measure service levels based on what they experience on their desktop. In most cases the performance monitors measure response time as viewed by the system, and depending on the connection between the user's desktop and the system, this may not accurately represent response times perceived by users.

Response time measurement becomes particularly complex in distributed client-server applications, when the user measures response time for a complete transaction representing a unit of work which may require several interactions with their desktop and, potentially, some interaction between their desktop and one or more servers in the network. New techniques are being developed to accurately measure end-user response time in distributed environments. Until these techniques become generally available, the solution is to instrument the application code, or to correlate system measured response times and resource utilizations with user-perceived service levels.

Job turnaround also presents difficulties for performance monitors. Again the most accurate measure here is to instrument the critical batch applications. If this is not possible, a global measure of job turnaround is usually available on commercially oriented systems, or you may be able to correlate job turnaround with resource availability for batch jobs.

Instrumenting the code is possible if the application has been developed in-house, and normally involves time stamping when each critical transaction or batch job commences and completes. This can then be summarized and periodically written into the performance logfile.

It is important when identifying exception conditions to ensure that false alarms aren't generated due to transient "spikes" in performance. Therefore we need to include a time factor in our exception finding logic. This time factor can be calculated from the service level objective by looking at an average over a period of time such as five or fifteen minutes rather than examining a shorter time interval. For example if our service level objective says that response time should be less than 2 seconds 95% of the time, we shouldn't create an alarm if response time is 2.5 seconds during a one minute interval. However if response time remains at 2.5 seconds for ten minutes, then this would constitute an alarm condition.

To expedite problem resolution we should also assign a priority and severity with each exception condition. The priority reflects the importance of the application, transaction, user or system to the business, while the severity indicates the extent of the problem. These categories will help determine the order and urgency of addressing and correcting exception conditions. Applications are generally defined in a parameter file as a grouping of processes, or user sessions. The priority for each application could also be specified in this file. Severity would be dynamically assigned according to the extent of the deviation from our service level objectives or by the significance of the performance bottleneck as determined by resource utilization levels or a combination of bottleneck indicators.

The management-by-exception concept has been successfully employed by a number of integrated network management systems (INMS), such as HP's OpenView and IBM's NetView. Performance management applications for both single systems and distributed environments are now becoming available which also utilize management by exception. In some cases these are integrated with the network management facilities to provide an integrated view of the complete environment.

When selecting or building an enterprise performance management application, it is important to ensure the application is both extensible and open. Most organizations have multiple hardware vendors, and multiple platforms from each vendor. Because of this the performance management application has to be able to monitor and manage these heterogeneous systems. It is unlikely that a single vendor will be able to supply performance monitors for every platform within your enterprise. Therefore the enterprise performance management application needs to provide a framework with the capability to integrate multiple vendors' performance monitors and system-specific performance management applications. In this way, there is a single view for all the exception conditions within your environment. When you have isolated a problem and wish to conduct detailed analysis, you can use the most appropriate system-specific performance monitor from within the same user interface. Another requirement is openness of data. You may wish to conduct custom performance or capacity analysis, and hence open access to the data collected by the performance monitors can be important.

Acting on an Exception Condition

Once we have identified an exception condition, we need to determine the cause and take corrective action. The first step is to isolate the problem to a particular system and, if possible, application. We then need to characterize the nature of the problem and determine which processes, users and resources (either system or network) are affected.

The complexity within this isolation phase will be determined by the computing facility's configuration. A problem within a large, complex, distributed heterogeneous environment may take considerable effort to characterize and isolate, particularly if client-server applications are involved.

There is usually a logical approach that most experts will take when resolving a performance problem. This approach can be captured in the form of rules which can be defined and executed in a logical sequence in order to automate the first level of problem isolation, characterization and resolution.

An example of isolating an exception condition is:

- Are there one or multiple exception conditions within the environment?
- Are multiple systems involved?
- Is it one application, inter-related applications, or completely independent applications?
- If multiple exceptions are related, is there one system acting as a server which is the primary cause of these alarms?
- What are the priorities and severities of each of the primary alarm conditions?

The intent of the above is to isolate each of the primary alarm conditions and to determine the order in which each exception condition will be addressed. Each primary exception condition can then be addressed by examining the alarm condition details including the service levels for the application (if the alarm was triggered by an application), and performance measures for the system resources. The goal is to identify which system resources are the current performance bottleneck, which processes are most heavily using these critical resources, and which processes (and hence applications) are affected by the performance problem. This should then allow us to explain why the exception condition occurred, and to make recommendations to correct the condition including the processes and system resources requiring corrective action.

General performance "rules of thumb" are available for most systems, from either the hardware or application software vendor, or third party consultants. These rules can assist in determining which resources comprise the bottleneck, however it is important to realize that to be most effective, the rules should be tailored to your own particular environment. The ability to tailor the rules will come with experience built by continuously monitoring the performance patterns of your environment under your applications and workloads. In a similar fashion, performance rules can also be ascertained for your network devices and topology.

These performance rules will allow the critical resources or system bottleneck to be identified for the particular exception condition under scrutiny. Individual processes can be examined to determine which processes are the heaviest users of the critical resources. Then, based on the relative importance of the processes involved (both the heavy resource users, and those experiencing performance degradation), corrective action can be specified. The corrective action could involve altering process priorities, or perhaps even cancelling and rescheduling a particular process. Note that the immediate concern is to relieve the current condition, but there may also be other longer-term actions required, such as increasing system resource capacity, balancing workloads, or reducing workload demands by altering job schedules or tuning the application itself.

The logical problem resolution sequence and the rules to determine critical resources, heavy resource consumers and impacted processes can be documented and automated within a number of available products which offer an "expert advisor" capability. It is also possible to develop a performance management application in-house using these concepts and techniques.

Automating the first level performance diagnosis is very rewarding even if you are fortunate enough to have available performance expertise. The expert will be able to focus on the more difficult situations, as well as implementing performance and capacity management tasks associated with becoming more proactive. These include identifying and resolving potential problems before users are affected as well as strategic and tactical capacity planning.

Becoming Proactive about Performance Management

In the above, we have defined a process that starts at defining our desired service levels, identifying violations of these, and then following a logical procedure to isolate the cause and take corrective action. However, because we identify the problem only after service levels aren't met, user productivity will already be affected. One of our objectives is to ensure that user productivity is not negatively affected, so we need to improve on the techniques outlined above.

We could set our service level thresholds lower, in order to provide a "buffer" between our alarm notification and any impact on user productivity. Depending upon the performance characteristics of the system or network, this could mean providing additional excess capacity which would never be utilized, and hence we wouldn't meet our second objective of optimizing the return on investment in computing facilities.

There are a number of alternative techniques, based on management-by-exception, which can more effectively identify potential problem situations. Rather than simply examining the current service levels and resource utilizations against fixed thresholds, we could build and continuously maintain a performance profile by hour of day and day of week. We would then monitor for significant deviations from the profile, which provides a more accurate representation of exception conditions.

Monitoring service level trends over time would allow us to characterize and quantify performance degradation trends and therefore project when service levels will no longer be acceptable. This could be accomplished by retaining the performance measures in logfiles, and then periodically using time series analysis to identify and project trends. Exception conditions would be based on exceeding our service level objectives within a specified future time period.

In order to accurately identify potential problems, we will probably need to use all of the above methods, together with examining a number of system performance and resource utilization metrics. For example, if we determined that we were approaching our service level objectives and that the critical resource was the system CPU, we would want to examine which processes were heavily using the CPU, which were CPU starved, how many processes were queued up waiting on the CPU, the relative priorities of these processes, and their relative importance to the business. The fact that we are approaching our service level objectives and that CPU utilization is high might not really present a problem if the primary CPU user is a batch process whose priority is lower than our important interactive applications.

Over-utilized system and network resource will be identified when utilization thresholds are exceeded. These resources could then be examined in more detail to determine if workload balancing would relieve the situation or if additional capacity is required. To do this we will probably need access to longer histories in order to completely characterize the resource utilization patterns. This information is available from resource management applications such as HP LaserRX. Similarly, we may also want to identify under-utilized resources within our environment which could perhaps be redeployed to support more critical applications.

It is possible to automate much of the above logic, which further relieves our performance experts, and will allow them to become even more proactive by conducting strategic capacity planning exercises. This will support our third objective for performance management, that of ensuring sufficient capacity within our computing facilities to meet current and future business demands. We may also want to review our management organization to ascertain if the use of these new concepts has any implication on the organization structure.

Organizing for Performance Management Efficiency

The capabilities outlined above facilitate segmenting performance management into two categories:

- On-line performance management to ensure user productivity is not impacted by performance problems within the existing environment, and

- Detailed analysis and planning to ensure system resources are utilized effectively and to provide sufficient computing capacity to meet business requirements.

The concepts and techniques outlined for management by exception and automated rules-based advice would allow a help desk to provide much of the on-line performance management. With closer integration between network, system and performance management, made possible by newly available integrated management products, the help desk could also provide first level network and system management.

This allows the available performance and capacity management expertise to concentrate more on ensuring that the computing facilities can support business goals, rather than being consumed by continual reactive problem solving.

Summary

Performance management can be more efficient and less labor and expertise intensive. The concepts of management by exception and automated advice can be readily applied to performance management, and there are products available in the marketplace which utilize these techniques. When selecting or building an enterprise performance management application, ensure the application is both extensible to cover all of your hardware platforms and open to allow custom analysis of collected performance information.

Implementing management by exception requires you to determine your desired state, which can be best accomplished by defining your service level objectives. In some cases, measuring against these objectives may not be simple, and may require you to instrument application code, or to identify correlations between resource utilizations and user-perceived service levels. The thresholds used to determine exception conditions should be specific to your computing environment, applications, workloads and business priorities. The concept of management-by-exception can be taken further to identify potential future problems as well as current performance problems. This provides a proactive approach to performance management, and will help ensure problems are identified and corrected before user productivity is affected.

Once problems are identified, pre-defined rules can be evaluated for first-level problem diagnosis and resolution. Again, these rules should be tailored to your specific environment. A starting point can be to use general performance rules provided by vendors or consultants, and to customize these as your experience with your environment grows.

Using these concepts and techniques can free your performance experts to conduct more detailed trend analysis and capacity planning, which means the capacity and capabilities of your computing facility can be more aligned with the business objectives of your organization. This allows all three objectives for performance management to be met.

* UNIX is a registered trademark of UNIX System Laboratories Inc. in the U.S.A. and other countries.

References

[McBRIDE] McBride, Doug, *"Service Level Agreements: A Path to Effective System Management"*, HP Professional, August 1990

[MILLER] Miller, George W. (Bill), *"Service Level Agreements: Good Fences Make Good Neighbours"*, Conference Proceedings of the Computer Measurement Group (December 1987), pp. 553-557

[MORRIS] Morris, Wayne S., *"Performance Management in a Distributed Computing Environment"*, Conference Proceedings of the Computer Measurement Group (December 1991), pp. 620-625

[ROSE] Rose, Marshall T., *"The Simple Book An Introduction to Management of TCP/IP based Internets"*, Prentice Hall, 1991

PAPER NUMBER: 5801

WIRELESS DATA COMMUNICATIONS: A PRIMER FOR RADIO FREQUENCY APPLICATIONS

Randal L. Shewmaker, CDP,CSP
Boeing Computer Services
P.O. BOX 7730 M/S K79-40
Wichita, KS 67277-7730
(316) 523-0395

ABSTRACT

Survival in today's business environment places greater emphasis on the concept of On-line Transaction Processing (OLTP) and performing data entry at the point of the transaction. Radio Frequency portable data entry products represent intelligent terminals capable of providing operators on foot or in vehicles with telecommunications capabilities similar to those enjoyed by management and clerical staff located in offices.

This paper will demonstrate that a well conceived application using Radio Frequency portable terminals and a host computer can provide sophisticated customer support in a manner equally effective as one using fixed video terminals.

MANAGEMENT'S CHALLENGE

The astonishing advances in transportation and communications technology during the past three decades have made the world smaller and its resources more accessible. New patterns of social, political, and business intercourse have emerged to form a new world order. We have become a "Global Village." Within the confines of that village, both governments and businesses have intensified their competition for a share of the world's markets.

For an enterprise to achieve and maintain "World Class" status, pressures continually mount to increase productivity and efficiencies while reducing costs. Customers have become more sophisticated and no longer accept delays in shipments, substandard quality, or poor service. The ever-present threat of competition requires a commitment to increase the customer's perception of quality because every company has an obligation to its shareholders to not only survive, but also to counter competition and pursue growth.

American management can no longer be content with competing on financial strategies alone but must learn to compete on a price, quality, and added value basis. The key to competitive success is a commitment to long-term objectives which provide a basis for continuous, measurable, incremental improvement. No sizeable firm can exist for more than a short time if its only objective is to maximize profits. As Mark Twain said, "Even if you're on the right track, you'll get run over if you just sit there."

As so often happens, two proven, yet here-to-fore unrelated technologies have been fused, by the catalyst of miniaturization, to place a new technology in the arsenal of alternative systems' solutions. Both radio and digital electronics have existed for many years. Now, with their technologies merged to form Radio Frequency Data Communications (RF/DC), the ability now exists to move away from systems designed to detect errors after they happen toward systems designed to prevent errors by identifying mistakes as they happen. RF/DC is indeed the "Wave" of the future.

RF/DC OVERVIEW

Radio Frequency Data Communications is the technology which allows portable terminals to communicate, in real-time (when the Enter key is touched), to a host computer without the need for traditional hard-wired transmission lines. RF/DC is used to locate terminals in places where it would be impossible or impractical to install transmission lines or where workers must have real-time access to a computer while moving about a large area. This is done by transmitting data through the air using radio waves.

Other than the transmission media, RF/DC is based on the same concepts as other real-time client/server networks. The hardware, of course, is different. RF/DC utilizes low power radio transmissions (.5 to 25.0 watt) to bring local area networks (LANs) to areas as small as a single warehouse or to an area the size of a campus. Three components are required for a functioning RF/DC system:

- a remote Radio Frequency (RF) terminal
- a radio base station
- a host computer

Portable terminals are comprised of a digital board, modem, radio transmitter/receiver, and an input/output interface. Most portable terminals have from two to sixteen lines of liquid crystal display, a keyboard, and a port for an optional bar code scanning device. These Portable Data Entry Terminals, also known as PDET's, can be permanently fixed at a location but are most often hand-held or mounted on a forklift or sideloader.

The radio base station receives the radio signals sent from the remote terminal and passes them to the host computer. The base station then broadcasts the information sent from the host out to the remote terminals. An RF network is a single point (base station) to multi-point (remote terminals) architecture. The base station also manages the communications on the network, serves as a protocol converter, and handles the interface to the host.

The host, of course, is where the application resides and can be anything from a dedicated PC or server, to a mid-range or mainframe computer.

TYPICAL RADIO FREQUENCY APPLICATIONS

In the universe of potential applications which could benefit from RF, several applications promise immediate hard savings, increased operating efficiencies, and continuing Return On Investment. Even though warehousing, manufacturing, and retail applications appear to be unique areas, they possess many similar processes which require mobility or require data to be collected where it is created. Most would also benefit from paperless systems because paper is an additional inventory item, paper systems delay material movement, and paper processing causes errors.

Shelf price auditing in the retail sales environment came about as a direct need resulting from the grocery store checkstand point-of-sale devices which have recently become so popular. Removing prices from items makes it imperative that the shelf price agree with the item price file. If it does not, the customer will be overcharged and the customer will be lost, or the customer will be undercharged and the profit will be lost. Many batch solutions are commonly in use today but they provide only partial accuracy. Without the ability to check the price file in real-time, there is no way of knowing what the price really is right now. It is impossible to be totally certain of a price if all one has to work with is a listing or a batch file download which was produced the day before. With mobile RF terminals, the shelf tag can be scanned, the product number transmitted to a host, and an immediate price audit can be returned to the portable in real-time.

The portable point-of-sale concept is an emerging requirement in the retail business. By using RF, each customer can have his own personal shopping assistant. Sales personnel can escort customers to desired items in a store, assist the customer in selecting the correct merchandise, and accept a charge card to pay for the items on the spot. With a personal shopping assistant, there should be no more long checkout lines, the customer would always be escorted, and occurrences of returned merchandise should be reduced.

In manufacturing applications, tracking work-in-process has a high potential for improved quality control and increased customer service. With a typical paper burdened real-time MRP II system, "you can't move something if you haven't received it. If you can't move it, you can't issue it to a job order. If you can't issue it to a job order, you can't build it." What usually happens with this type of "state-of-the-art" system, is that product is built and shipped out the door in spite of the system. A phantom system known only to the workers gets the product manufactured, usually before the initial receiving document is ever keypunched. With an RF "paper-least" system, these accounting processes can be captured as they happen, in a timely, correct, and accurate manner.

Warehousing solutions have traditionally been among the first applications implemented using RF/DC. The use of RF in receiving areas allows the arrival of new shipments to be posted to a data base as soon as they arrive at the dock. Checking the incoming material against an open purchase order while it is still on the truck, allows incorrect shipments to be refused before they are unloaded. After all, the first rule of materials handling is "don't handle it." Why should product be unloaded and accepted into stock only to later find the error? The stock then gets handled at least one more time, the company gets billed for something it did not order, and the ordeal of following the supplier's return process to receive a credit is endured.

By instantly recording an incoming shipment, the goods are now available for immediate use instead of waiting twenty-four hours for receiving documents to be key punched. If a company processes $100,000 of goods per day and it takes twenty-four hours to post sales and receipts, then there is $200,000 worth of excess inventory on the books all the time.

In some instances, the incoming stock is on backorder. Why should the stock be put away only to pick it again tomorrow? The RF system should instruct the storekeeper whether to put the stock away. With this capability, the backorder could be filled forty-eight hours faster than with a paper driven receiving system.

Direct putaway programs using RF consider space limitations within the warehouse. This allows the computer to direct the warehouse worker to place incoming stock in the best bin based upon some predetermined criteria such as placement of fast moving stock in areas where it is most accessible and nearest the shipping point.

RF/DC seems to have been made for dynamic storage/random warehousing. With RF it makes no difference where something is put away. The storekeeper simply puts it away and lets the RF system remember where he put it. Basically, the storekeeper scans a bar code on the pallet and a barcode on the bin location where the stock was placed. Later, when the material must be retrieved, the storekeeper inquires on his portable RF terminal and lets the system tell him where the product is. This application could even tell him if stock is really available, therefore preventing a deadhead trip which would be costly both in wasted time and wear on equipment. The system could also suggest, based upon some method of stock rotation, which product must be picked when it is stored in multiple locations.

In terms of controlling inventory within a warehouse, RF is the technology of choice. In a warehouse it would not be unusual for several storekeepers to be doing picking, while several others are doing putaway, and several clerks are doing cycle counting. In such a dynamic environment, it would be virtually impossible to have total accuracy in inventory. Yet, with RF and the appropriate hardware and software, the cycle counter can be only a foot away from the storekeeper doing putaway and still get an on-time, correct response to the accuracy, or inaccuracy, of his count. Of course, the best time to do a cycle count is when the quantities are small and someone is there. If the activity level in the warehouse could tolerate it, the possibility could exist for the RF application to prompt the storekeeper to perform a cycle count immediately after completing his putaway or picking operation.

COMMUNICATIONS

Generally, the two most common types of RF transmission multiplexing are polling and contention. Both are roughly based on the CCITT X.25 recommendation for packet switching network access. Most also use cyclical redundancy checking (CRC- 16) for error correction.

In a polling system, the remote terminals must wait to be authorized (polled) to send an inquiry to the host. The base station/controller maintains a list of terminals to be polled and usually possesses logic to alter the polling sequence to poll less frequently those terminals which have been inactive for a period of time but remain logged on. This grants more priority and air time to the terminals which appear to be busiest.

Polling systems require the base station to be on the air at all times performing either broadcasting and polling routines or receiving data and acknowledging inquiries. There is a possibility this continuous activity could create interference on other radio systems on adjacent channels if they were used nearby.

Polling systems, by their high overhead nature, provide lower terminal response times and can best serve an environment where a limited number of terminals are in use and where relatively high transaction or data volumes are required.

Contention (Carrier Sense Multiple Access [CSMA]) systems are usually better suited for applications which require a large number of remote terminals and where transactions are random, brief, and infrequent. Before transmitting, each remote terminal listens for a carrier wave from another terminal which indicates a transmission is in progress. When no carriers are sensed, the terminal assumes it is clear to transmit its request. Because the terminals do not listen for the other terminal's data, only their carrier, and because the carrier can be heard at much greater distances than can the data, CSMA protocols can service larger areas than can polling systems.

Half-duplex systems use two frequencies. The base station listens on one frequency and transmits on the other. While the terminals also utilize two frequencies, they are unlike the base which can transmit and receive at the same time. The terminals can perform only one function at a time. Even though it is possible to gain a bit of a performance advantage with a half-duplex system, simplex, single frequency systems are most commonly used.

Almost any interface between the base controller and host can be acquired with the standard RS-232C asynchronus ASCII being the most widely accepted. As previously mentioned, most base stations are preconfigured with a packet assembler/disassembler (PAD) protocol converter to transform the data packets into the asynchronus character strings used by terminals. The hardware vendor should provide any host driver packages, terminal emulation, or other protocol conversion device needed.

NARROW BAND TECHNOLOGIES

Citizen band radios, marine radios, car radios, and the radios used by hobbyists to fly model airplanes all operate on what is known as "narrow band". Even garage door openers and cordless telephones operate on narrow band. Narrow Band is so called because the available band width is only a few kilo-Hertz (KHZ) wide. All electromagnetic radiation, including radio, is measured in cycles per second (CPS) and CPS are represented in Hertz.

The Federal Communications Commission (FCC) restricts all narrow band radios to a 12.5 KHZ bandwidth. This restriction, coupled with deficiencies in signal quality inherent with current radio technology, greatly impacts the performance of RF data systems as we know them today. Even by using all the available bandwidth, the maximum data rate that is feasible on a narrow band channel is 9600 Bits Per Second (BPS). Most RF systems sold today have a much greater tolerance for error by operating at 4800 BPS and below. Beware of the salesman who insists that his 4800 BPS system offers twice the performance of another vendor's 2400 BPS system. Throughput, not data rate, is the performance gauge of a system.

All RF transactions must sequence through a series of steps before they can be completed. How each step is handled and how long each step takes to complete vary with each vendor. These steps are:

- Listening to be polled or if the channel is clear
- Raising the carrier
- Sending the data to the base station and on to the host
- Waiting for a response from the host back to the base station

- Listening for an incoming inquiry
- Raising the carrier
- Sending the data back to the terminal

The amount of time it takes for a terminal to become ready to transmit data from an off state is called "turn-on" time. Some vendors refer to "turn-around" time as the time that is required to cycle from a receive state to a transmit state. These two definitions usually yield the same measurement. Just as a fluorescent light requires more time to turn-on and stabilize than does an incandescent light, the turn-on/turn-around time of many vendor's products can vary as dramatically.

Consider two similar radios, both requiring 200 milliseconds turn-on time, but one with a data rate of 2400 BPS and the other with a 4800 BPS rate. At 2400 BPS, data is transmitted at a rate of one character every four milliseconds or one character every two milliseconds at 4800 BPS. Therefore, a fifty character data string would require two hundred milliseconds to transmit at 2400 BPS and one hundred milliseconds at 4800 BPS. If the host to base station delay is three hundred milliseconds in each case, throughput times on the 2400 BPS unit would be 1100 milliseconds and 900 milliseconds on the 4800 BPS unit.

Transmission Phase	2400 BPS Unit	4800 BPS Unit	2400 BPS Unit
Terminal Turn-on	200 ms	200 ms	50 ms
Send Data	200 ms	100 ms	200 ms
Host Delay	300 ms	300 ms	300 ms
Base Turn-on	200 ms	200 ms	50 ms
Send Data	200 ms	100 ms	200 ms
	------	------	------
Total Throughput	1100 ms	900 ms	800 ms

Figure 1

If the 2400 BPS unit were upgraded with a faster radio giving 50 millisecond turn-on times, a throughput rate of 800 milliseconds would be realized even at half the data rate of the 4800 BPS unit. With current technologies, from 14 to 29 percent of on air time is spent bringing the hardware up to a steady state.

Power, range, receiver sensitivity, data rate, outside interference, and optimal antenna placement are all worthy of consideration when selecting an RF/DC system. Negotiating the maze of spec-sheet data also makes the selection process quite difficult.

There are almost as many approaches to the implementation of narrow band systems as there are major vendors. Some prefer to use a single high-power base station to broadcast out to the remote terminals and a group of remote receivers to get the terminal's signal back to the base. Since radio waves behave in a similar manner as light waves, this approach creates a possibility of shadows.

The cellular approach to narrow band RF/DC has become popular among some other vendors and allows terminals to move throughout a large area of coverage. The network software will transfer the control of the terminal from one cell to another based upon the terminal's relative signal strength with respect to the fixed-position base receivers. The controlling software is quite complex, but the concept has been proven, and many successful implementations exist.

Multiple antenna options use a single base station and a signal splitter to route the radio signal to various antenna locations. However, the most elegant and, of course, the most expensive approach to narrow band RF/DC incorporates a distributed antenna. This approach effectively creates an antenna from a length of wire which loops and meanders at regular intervals throughout the coverage area. The wire is spaced so that a portable terminal would never be more that fifty feet from the next length in the loop in any direction. A distributed antenna is usually installed during the construction of a building where the antenna can be embedded in the floor. Distributed antennas are appropriate in buildings with moveable walls since the antenna would never need to be relocated.

Among these four approaches to narrow band RF/DC, there is no preferred technique. All the approaches have merit and can work effectively if they are engineered correctly to ensure that the site is completely covered. Any reputable vendor will guarantee coverage of a prospective area if they are allowed to perform a site survey and provide direction in terms of power and antenna placement.

SPREAD SPECTRUM TECHNOLOGY

The spectrum of radio frequencies is a finite resource tightly controlled by the Federal Communications Commission. Spread Spectrum, or broad band, provides a very low power, secure, interference-free network which does not require an FCC operating license.

Most metropolitan areas have very few frequencies in the narrow band (400 mega-Hertz [Mhz] and 800 Mhz) range available for licensing. Interference issues are becoming paramount, particularly in the retail sales environment, where a one watt narrow band radio can cover a two mile radius. The potential impact of a narrow band RF application in use in every store in a shopping mall or a large commercial district could be staggering. To address this issue, the FCC has reorganized and reallocated some of its narrow bands and made the "garbage bands" (900 Mhz to 928 Mhz, 2400 Mhz, and 5725

Mhz) available for unlicensed public use. These newly acquired bands, coupled with spread spectrum technology acquired via the Freedom of Information Act, have enabled Spread Spectrum RF/DC to be commercially marketed.

The spread spectrum concept was originally implemented by the military more than twenty years ago to facilitate communications between aircraft and ground support troops on the battle field. The low-power, redundant properties of spread spectrum made it impossible for hostile forces to jam or read communiques. Spread spectrum is an alternative technology which is appropriate for environments where there are high populations of terminals or where high throughput requirements exceed the feasibility of narrow band solutions.

With spread spectrum, data is encoded and transmitted at very high rates of speed. The broadcast is repeated numerous times over constantly changing frequencies based upon some pre-established algorithm which can be configured differently for each installation. Essentially, the data is duplicated, or "spread" over the entire available spectrum effectively flooding all frequencies within the band width with the data. When the data is intercepted, all iterations of the transmission are compared, and the data string most likely to be correct is sent on for decoding. Data rates six to twenty-six times faster than narrow band (60 KBPS to 256 KBPS) are common. The radios have turn-around times of 3 milliseconds instead of the 40 milliseconds found in the best narrow band radios.

The laws of physics dictate that the higher the data rate, the shorter the broadcast range. Powered at less than 1 watt, spread spectrum radios can only broadcast a distance of 500 to 1000 feet. To cover a larger area, additional base stations must be installed to implement a true cellular network. CSMA protocols are used to eliminate the overhead associated with polling methods. Yet, the overhead associated with the speed of the transmissions, the encoding and decoding, the complicated error checking and, the cellular "pass-off" algorithm processing is beyond the capabilities of a base station. Spread spectrum applications require a front end processor to perform network management and hashing functions in conjunction with the base station.

TERMINAL CONFIGURATIONS

Radio frequency terminals can be considered portable, battery-powered, tele-transaction computers which collect, store, process, transmit, and, if desired, print data. Some are simply dumb conversational terminals which store simple edits while others are as complex and capable as a desktop computer.

Three basic configurations fill every application niche. Tethered terminals represent those units commonly found installed in fork lifts, side loaders, and other types of delivery vehicles. The terminal and radio are separate units connected by a cable. The radio is interfaced to the terminal via a modem. The power sources for these units are usually separate, and some draw their power from the vehicle's electrical system. Since these high-performance systems were never designed to be hand-carried, weight is not a consideration, and more space is available for complex radio electronics.

When light-weight, maximum portability solutions are required, integrated terminals are most commonly selected. Integrated terminals make compromises with the radio electronics to incorporate both the terminal and the radio in one case. These terminals are usually hand-carried and have a scanner attached, but some vehicle- mount designs do exist.

The most recent innovation in portable terminals has been to expand the integration concept to include the scanning device. These light-weight, ergonomically designed configurations are quickly gaining general acceptance. Their one-handed design eliminates the need to juggle both the terminal and a tethered scanning device when performing some operations. Terminals integrated with scanners are available with both narrow band and spread spectrum architectures.

The range of terminal weights varies considerably, even throughout a single vendors' product line. Gross terminal weight is usually dictated by the weight of the accessories and options installed on the terminal. Additional memory, batch storage capabilities, and long-life batteries affect the terminal's over-all weight. Most integrated terminals weigh less than two pounds.

Since the terminals are actually hand-held computers, they require operating systems. Until recently, these operating systems have been proprietary in nature.
+Now vendors are becoming responsive to the need for standard, portable, programming environments and have made MS-DOS available. With MS-DOS, almost any application language can be used. Consider a portable terminal running multiple applications under Windows.

Applications software is developed on a desktop computer (PC) and is usually down-loaded (soft-loaded) to the portable terminal where the program will reside on an electronic virtual disc. The major disadvantage of soft-loading an application is the relative volatility of the program in the event of total battery failure. The alternative is to burn the application on an EPROM and plug the program into the terminal. The recent introduction of EEPROMs (Electronically Erasable Programmable Read Only Memory) has made it unnecessary to remove PROMs from the terminals to change their coding. Application programs and forms should always be stored in the terminal and never down-loaded via the RF link at the beginning of the work day.

To simplify the development of custom applications, vendors typically supply application development utilities and callable library routines called "programmer's development kits". These library routines include the programs which actually effect the on-air work and other utilities which are used to design screens for the terminals. Programmers typically develop applications, using Turbo C or other compiled language, to accept input, perform edits, form a record to be transmitted, and call the vendor's library routine to make the transmission and to process the message returned from the host computer.

Some terminals have the ability to store data internally in the event the radio link to the host is lost. Depending on the application, this store and forward technique could allow the data to be transmitted, in batch, to the host at a later time.

Except for a few mobile units mounted in vehicles, power for the portable terminals is derived from an internal battery pack. Due to their exceptionally flat discharge curve, Nickel Cadmium (NI-CAD) batteries are commonly used for general purpose operation. Miniature lithium batteries protect main memory and soft-loaded programs when main battery packs are changed.

Depending on how frequently the scanning device is used and the number of transmissions made, one set of fully charged batteries should last for ten hours or, at least, one shift. High wattage radio transmitters will consume battery power at a higher rate than lower powered radios.

Trickle chargers require twelve to sixteen hours to bring a pack of Ni-Cads to full power, but they tend to encourage chemical memory within the batteries. If a battery is used for only one hour per day and is recharged every day, the battery will eventually retain only one hour of charge.

RAM chargers provide a quick (two hour) charge in the event of an emergency, but they tend to overheat the batteries causing them to vent their electrolytes.

The best device for long-term care of batteries is a battery cycler. This device drains the batteries just to the point where polarity could reverse, jolts the battery with a heavy charge to remove the chemical memory and ensure a flat discharge rate, and then trickle charges to full capacity. This process usually requires fourteen hours.

Battery management programs are essential to ensure the availability and reliability of the terminals and to ensure the continued success of the RF/DC project.

GENERAL PROJECT CONSIDERATIONS

As with any project, there is no substitute for planning and project management when installing an RF/DC system. It has been said that "a carelessly planned project will take three times longer to complete than expected. A carefully planned project will take only twice as long."

Those who are unfamiliar with RF technology, should start simply with a small application. Managers should not bet their credibility on their first try, but rather pilot their way to success. First RF/DC attempts should be directed toward interfacing RF with an existing system, not integrating systems with RF. A managers' first experience should be one from which he learns, not one from which he recovers.

An RF/DC system should be purchased on the basis of performance requirements, terminal populations, the size of the physical area to be covered by the radio, and, of course, the budget. Integrating a variety of heterogeneous equipment into a cohesive network forces the manager to face various potential compatibility issues. Vendor's specification sheets have allure and mistakes can be very expensive. Unless a manager really knows what he is doing, he should not buy an RF/DC system "by the pound", but rather purchase a complete set of hardware.

A single-vendor approach will facilitate the ability to isolate malfunctions and take appropriate action. A vendor should be selected in much the same way one would select a business partner. Several vendors should be interviewed and one should be selected on the basis of their reputation, their ability to meet requirements, and their ability to "fit in" with the project team. If a team player is chosen, the chances of success are greater and contract negotiations and problem resolution will be less awkward. Competition among RF vendors is so intense that managers can demand that they be experts, but there is no substitute for managers knowing what they need and having the ability to, at least, be conversant on the technology.

With the RF system requirements well defined, the vendor will address throughput issues like data rates, and hardware and protocol overhead. Managers, however, must assume the responsibility to provide an acceptable host response delay. If there is not sufficient computing power to do the project, the computer must be upgraded or the project should not be done. The RF network should not be overloaded with terminals. As with any other type of network, activity should be kept below forty percent of its maximum rated capacity. If the end-user perceives unacceptable response times or if he does not see that the system makes his job easier, it will be the kiss of death for the project. No matter how elegant the system may be, in the final analysis, only the end-user can assess how well the system meets his goals.

If at all possible, RF/DC hardware should be amortized over a three year period. RF/DC technology is rapidly advancing and growing in complexity. Uncertainties still exist in industry standards. These issues should be expected to quickly obsolete the RF equipment. Vendors will continue to support their equipment as long as it is installed but the equipment's useful life will be much shorter than it's physical life.

If a narrow band solution is selected, managers must be certain that a site license has been procured before purchasing any RF equipment. The FCC has a set of rules designed to ensure that radios on different channels do not interfere with each other and rules that prevent commercial radio users from interfering with the public at large. It would be unfortunate to purchase equipment that operates on a frequency that the FCC has previously authorized another company on the next block to use. The application process is very simple and costs less than $100.00. A license is valid for four years and is renewable indefinitely. Any reputable vendor will assist in filing for a site license.

It is just as important to have a program of spares management in place as it is to adhere to a data backup methodology because even the best equipment fails at some time. The number of spare scanners and terminals and the possibility of a spare base station should be thoroughly considered and their costs included in the project justification. The logistics involved in replacing and repairing defective equipment must be established. The repair and maintenance costs associated with spares management should be budgeted at the onset of the project.

Since each portable terminal is indeed portable, a system of accountability and configuration management is essential to ensure that all terminals are programmed with the same version (release) of your application software.

Three fundamental rules to reduce on-air time and increase response should be considered when developing an RF application: (1) If the terminal operator doesn't need to know it, don't transmit it. (2) If you can process it on the portable terminal, don't send it to the host. (3) If you have sent it once, don't send it again.

CONCLUSION

Users have become increasingly dependent on the products of technology and have taken for granted the tools and systems which have so profoundly affected the way they work. To remain responsive to their on-going demands for more effective business solutions, managers are under increasing pressure to become proficient in virtually all aspects of innovation.

The dizzying rate of technological change has made announcements of new and radically improved technologies occur with disquieting regularity. Data processing professionals must be on constant guard not to adopt new solutions and then frantically search for an appropriate problem. Even though the universe of available technological solutions is formidable and the learning process grueling, managers must never hesitate to adopt an appropriate, innovative technology if a legitimate, deserving problem is identified. As Henry Ford said, "If you need something and don't buy it, you'll eventually pay for it without having it."

Radio Frequency applications are no more difficult to develop and implement than any other real-time solution. While RF/DC technology is fascinating, it is important to remember that systems are developed to serve some business function, not to showcase a new technology. Often, there is a tendency to become immersed too quickly in the glamour of a new technology without considering how appropriate that technology may be for a specific application. Care must be exercised to select the right tool for the right job.

Even though extraordinary advances in computer technologies have far exceeded the ability to apply and manage them, managers must not become convinced that they are faced with nothing but problems; but rather opportunities. Henry Ford understood that "whether you think you can or whether you think you can't, you'll be right in either case."

Managing a Multi-CPU
Multi-Vendor Data Centre

David Woodhouse
Carolian Systems
Unit 5, 3397 American Drive
Mississauga, Ontario
CANADA, L4V 1T8
(416) 673-0400

Managing A Multi-CPU Multi-Vendor Data Centre

During this presentation, I will discuss

1) Day to day operations in a medium to large Data Centre.
2) Task Identification. Who is doing what and why.
3) Task Elimination and automation.
4) Data Processing Policies and Procedures.
5) Implementation of a "Lights Out" Data Centre.

Before I go any further, I would like to give you a quote from a 15th century "Management Consultant". This observation is one that will help you prepare for the task of moving from a traditional data processing environment to an automated one.

"There is nothing more difficult to take in hand, more perilous to conduct or more uncertain in its' success than to take the lead in the introduction of a new order of things."

Managing data centres, as they grow from a single CPU to multi-CPU, from a single site to a network of distributed machines and from a homogenous environment of HP 3000's to one that includes a variety of machines from different manufacturers running under several operating systems, is indeed a daunting task. To attempt to automate the management of this computing environment is to indeed "introduce a new order of things". It is worth noting that operations are essentially the same regardless of hardware platform or operating system.

The changes in your computing facilities did not happen overnight. You did not come in one morning to find that you had gone from a single 70 to four 967's with a few other odd boxes (DEC, Sequent etc) sitting in 2 or 3 data centres around the country. This sort of growth happens slowly and the typical response to such growth is simply to add people, train them to do what your existing people do and deploy them as the load and geography dictate. For a while that may work, but one of the interesting things about increasing the complexity of a system is that it tends to quickly demonstrate shortcomings in the way you do things. As I pointed out earlier, your computing environment changed gradually and automation must be a gradual process as well.

Before I go too much further, I should let you know what I consider to be the key operations areas.

They are:

> Performance Management
Being on top of the minute by minute processing load such that problems are detected as they occur and in the longer term, analysis of the change in the resource usage such that upgrades and enhancements can be planned for and dealt with in an orderly and controlled fashion.

> Report Management
The automated distribution of printed output based on the physical location of the report addressee and the hands-off control of secure printing of sensitive output. This must take place over the entire network.

> Job Control and Scheduling
Automated control of the batch processing environment such that the processing is completed in a timely manner while having minimal impact on on-line users.

> Media Management
Up to the minute tracking of all media (back up and original distributions of software) regardless of type. Where is it and what is on each volume?

> Console Monitoring and Systems Operating
The execution of appropriate, consistent and timely responses to all console message traffic.

> Security Management
Transparent control of system access which is restrictive without making it unnecessarily difficult for authorized users.

> Data Integrity Management
Prevention, and failing that, the detection and repair of broken chains and logically incomplete records.

Some of the above items are probably part of the daily routine in most shops, some receive little attention at all and a few are completely ignored until you are forced to react to a disaster. Lack of attention to some of the above items can lead to severe problems in an automated operations environment which could cause your systems to become unusable before you have time react.

Opportunity For Error

For the most part, as managers, you rely on operators to monitor the systems and take the correct action or to inform the appropriate person of problem situations as they arise. This portion of the paper is not about operator bashing. It is a statement of reality based on 15 years of experience in commercial and academic computing. In most companies, the entry level computing position is junior operator and the duties associated with this position usually include a large number of midnight shifts. Training is usually hurried because the sooner the new operator is up to speed, the sooner the trainer gets to switch back to the day shift. The next new operator is trained by the current junior operator in the same hurried manner and so on. This is a high stress position which is usually one of the lowest paid in the data processing area. This is the first opportunity for error.

Hardware manufacturers have one thing in common. They treat "THE CONSOLE" as the clearing house for everything. All hardware level messages go to the console. All operating system messages go to the console. Software developers, taking their cue from the hardware companies, send all utility and applications messages to the console. All these messages assume that there is a person there to receive the message and secondly that the person is trained to deal with the message. This is probably wrong on both counts. In most shops, operators are compensated and promoted on their ability to ignore messages. If it isn't a tape mount request or a printer related message, it is interference and should be treated as such. This is the second opportunity for error.

When an operator with little experience happens to see a message which is not routine, he can always refer to the policies and procedures manual which is prominently displayed in the computer room. This document is current and complete and details the correct response for every system event. Unfortunately not every data processing department has a complete and current policies and procedures manual. This is the third opportunity for error.One current trend which has had an interesting effect on the operators job is that of consolidation. Many shops are reducing the total number of machines that they have by migrating to much more powerful systems. An HP 3000 Series 980-200 is theoretically capable of supporting 1250 concurrent sessions and represents the computing power of ten to twelve Series 70's. It is a rare shop that would consider running ten 70's with one operator, but now, a single operator gets the task. There is only one console to be concerned with, but there is 10 times the message traffic. During busy periods, there is the constant problem of critical messages scrolling off the screen without ever being noted.

If you get nothing else from this paper, I want you to carry this idea with you. **Your systems will be managed. Whether they are managed by policy or by default is up to you!** Without a considered set of policies which are current, clear and accessible to everyone, the default is what you are going to be left with. The result will be that things as basic as group and account structure will vary from machine to machine. Security will be implemented in a non-standard way. Even back-up timing and procedures may be different on various machines within a network of machines. In order to get a clear handle on the existing procedures in your shop, whether you have policies and procedures in place or not, is to analyze the console logs. This will tell you very accurately, though not without considerable work, how the systems are being managed and what the perceived correct response to routine system events happens to be. Unless those responses are predetermined by management and implemented through thorough and regular training, you will find that they vary from machine to machine and from shift to shift. This is not a particularly desirable situation.

Opportunity for Improvement

There are four ways in which you can make substantial improvements in the way in which your machines are operated. These are:

> Task Simplification
  Make it clear what action is to be taken in response to every system event. This must include those instances where the appropriate response is to wake someone up in the middle of the night.

> Improve the Interface
Through the use of operations automation software, make it easier for operations staff to identify those messages that require their attention. This means the suppression of trivial messages and the re-distribution of critical messages to the individual best able to deal with it.

> Task Elimination
Once you have created a document that describes the appropriate action to be taken for routine system events, convert that statement to software and let the computer manage itself. A prime example of this is the Batch Scheduler.

> Automation
As research identifies more and more tasks which are repetitive in nature, events to which the response is fixed or adheres to a set of rules, more and more of the operation of computer systems can be automated. Achieving a lights out computing may not be possible or even desirable, but a "dimly lit" environment can be a reality in a short period of time.

Automated Operations

The decision to automate the operation of a computing facility must be made at the highest level within an organization. This is true for a number of reasons. The first is the commitment of time on the part of people who understand the company's business as well as the company's computing environment. The first phase of the automation project involves a great deal of research. Internally, you must discover how things are done now and how they could be done better. Externally, you must investigate the marketplace to see what solutions are available to meet your needs. If none seems suitable, you may decide to create your own solution or abandon the project for a time. Next, you must assess the impact of automation on the organization. Such an undertaking will have costs as well as cost savings associated with it. The expenditure of large sums of money and perhaps the elimination of some staff are decisions that must be taken by senior management. There are also political aspects to this sort of project. The reduction or redeployment of staff can have an effect on departmental budgets. Shifting control from many locations to a single location may cause some to feel that they are loosing their power over the machine. With change comes a certain amount of uncertainty that some may engender resentment or fear.

Having an understanding of the negative aspects of such a project, why would anyone undertake to automate a computing facility? The answer is twofold. The first relates to cost savings. Quite simply, the more tasks you can automate, the fewer there are left for people to do. This means you require fewer people to do the same work. Cost containment and increased efficiency in all areas are what will allow your company to remain in business and to become increasingly profitable. The second relates to growth. As your company grows and requires additional computing power, you simply copy the automated procedures on to the new systems. You don't need to add staff.

The implementation of automation, just like your need for such automation must be gradual and carefully planned. There are obvious places to start where the incremental improvements in efficiency will be greatest and the disruption to routine the easiest to deal with. As a vendor, I can give you one piece of advice that may save you and several vendors a great deal of time. Before you talk to a vendor, before you look at a data sheet, take the time to talk to the users. Build a shopping list of must-have and nice-to-have features and assign a value to each. If possible, come up with a dollar figure that represents what the automation of a task is worth. You can then embark on your investigation of the marketplace with a clear idea of what you need, what you want and what it is worth to you. Products can then be selected or eliminated based on features and/or price. That up front investment in needs analysis will save you time by reducing the number of products you consider and will also insure a better fit in the user community.

The place to start the automation project is with the console logs. You must understand the flow of information from your machines to the consoles. Once you have a clear understanding of the nature and volume of the message traffic, you must determine how each of the messages, both routine and obscure, is dealt with by the operations staff. What does the operator deal with directly, what gets referred to a systems specialist, what messages get passed to programmers and what gets ignored. At the end of this exercise, you will have a chart which details the flow of operational information through your organization. This will form the basis for a plan of "Rational Distribution of Information". This is a phrase that my company has been using for years to describe the concept of placing the right information in the right place at the right time.

As you further analyze the flow of message traffic, you will find that much of that traffic is informational in nature; that is, messages which requires no reply. It is often notification of the successful completion of a job, or the movement of a spoolfile. As you automate the operation of your computer facility, you have no need to review those types of messages. You will develop an "operations by exception" strategy which requires you to deal with only those system events which can not be dealt with by standard automation software such as job schedulers and spoolfile management tools.

Tools that support this operations by exception strategy are few and in some cases immature, but they do exist. When researching the marketplace, look for tools that can be implemented in a staged manner, that require a minimum of re-training, that are easy to customize and that have a clearly defined growth path that is consistent with the directions of your own.

Summary

To recap, let me repeat the following points:

Operations consists of:
- Performance Management
- Report Management
- Job Control and Scheduling
- Media Management
- Console Operations
- Security Management
- Data Integrity Management

Potential sources for problems in a computer facility appear to revolve around operators and is due by and large to:
- Lack of Training
- Lack of Experience
- High Stress/Low Pay
- Assumptions on the part of hardware and software developers and
- Increasing complexity of the computing environment.

Improvements can be made in the operations area through:

- Task Simplification
- Improving the interface
- Task Elimination and
- Automation.

An operations automation project must be supported at the highest level of management and undertaken with a full understanding of all the associated benefits and costs. The project must be done gradually based on a careful analysis of the flow of information through the operations organization. It is suggested that commercially available tools be used wherever possible.

Automation tools must allow you to turn out the lights in the computer room with absolute confidence. The tools must fit your implementation plan, be easy to use and learn, be customizable and fit with your future plans.

Following these guidelines, the automation of a data centre can be completed with a minimum of pain and disruption to your organization.

PAPER NO.: 6001

TITLE: The Department of Defense and EDI

AUTHOR: David Frenkel
EDI Products, Inc.
1715 West 31st Street
Minneapolis, MN 55408
612-827-9237

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

Architecture

Mapping Business Needs to Technology

Prepared for

Interex

Computing Management Symposium

Atlanta, Georgia
March 23-25, 1992

Presented by

James F. Dowling
Bose Corporation
Manager, Information Technology Planning

The narrative that lies below is intended to provide a reference for use after attending the presentation of the Architectural Approach. In consideration of those who will be carrying the proceedings and because of the rather small format of the proceedings document, the sixty or so figures and illustrations will only be provided to those who attend the presentation or to anyone who writes to me for a copy.

For a copy of the presentation slides, please send me a note with your return address to:

James F. Dowling
Bose Corporation
The Mountain
Framingham, Massachusetts 01701

Introduction

There are as many approaches to the design, construction and delivery of information systems as there are practitioners in the field. Each of us takes what we feel is the best from the various methods that we have been exposed to and we blend into it our personal feelings about what is right and what fits our current environment to form a mode of operation for our particular work group. Several key concepts drive our approach. They include: the culture of the company and work group into which the process will be incorporated; the methods that we have been exposed to; our personal experience with the various methods; and our desire to employ the best methods that are available at the time.

Over the past two years, I have been trying to bring a collection of "best in class" solutions and methods together into a cohesive and complementary package that can be used at all levels of the information technology delivery process. The program that I have been working on includes multiple divisions of our company. It includes the relationships among various Information Technology organizations including Manufacturing, Engineering and Business Systems. It includes operations in several countries. Most importantly, it takes into consideration the approach of Total Customer Satisfaction through Managed Expectations.

When Information Technology was introduced at Bose, there were two departments that were involved. The Engineering department was working on signal analysis and measurement systems and the Information Systems department was working on paper flows and Sales Analysis systems. The two groups communicated regularly but had no work related goals in common. When the Engineering department opened a project to perform Materials Requirements Planning for a large project; there seemed to be a conflict of goals between them and the Information Systems group that was looking to automate the Inventory Control system for the Manufacturing groups. In actuality, there was no conflict of goals, only a conflict of time frames and approaches to the solution.

The Engineering group implemented an on line interactive MRP II system while the Information Systems group built an automated Kardex system. The Engineering group was several years ahead of the industry while the Information Systems group was merely automating an already flawed approach to managing material inventories. When the I/S group came to review our success; I coined the motto for them to be "We Automate Problems." Only many years later did it occur to me that what was going on between Information Systems and Engineering and Manufacturing was a failure to clearly define processes and desired process improvements. Once the Process Improvement approach was adopted by me, a high degree of clarity was brought to the value of business process analysis versus systems analysis. This is the first major building block for an architectural approach to Information Technology delivery.

The title, Architect is protected by law to be used only by those who have demonstrated a certain set of talents and skills after fulfilling a requisite education program. When one employs the title, Architect, s/he is stating that consideration of form and function will be given within the context of prevailing law and standards in the development of designs for structure and site of a construction project. The key terms here are Form, what it looks and feels like; Function, how it fits to its purpose; Law, How it fits within its environment and how safe it is; and Standards; How it fits within the generally available set of tools, skills and materials.

The second major building block for the architectural approach is to include the holistic professional approach by incorporating Form, Function and Fit (replacing Law and Standards) thereby bounding the solution set to some extent.

At this point the approach reads something like this. When we identify a business process that demands improvement; we apply a set of methods to reveal the functions that will be addressed then we construct solutions by selecting from a set of standards and guidelines to assure a good fit into the overall Information technology solution set. We then use the imagination and creativity of the client and the information technology professionals to build a solution in the appropriate form for the client.

Let's refer back to the Architect to see how s/he manages the client's expectations. The client is urged and assisted to define exactly what is expected of the structure. How many bedrooms, baths, and automobiles define some broad requirements. Special spaces such as office, library, sewing room define extra-ordinary requirements. Shapes, colors and light define very specific requirements that will often override the basic requirements of hot water and safe stairways when it comes to evaluating total client satisfaction.

The Architect will produce sketches, models and now even virtual reality computer models to demonstrate the proposed configurations and to extract the feelings as well as the needs. One of our Information Technology innovators at Bose recently noted that with some new technologies that he has been working with, he can now "Give his clients both what they want and what they need." This is another of the key components of the architectural approach. We need to develop from the client the specifications of what is desired and the importance of each function. We then must add what we know from our professional experience must be included to ensure that the function and features will continuously meet the clients needs. This ensures that the solution can be maintained and funded by both the client and the support staff over the life of the solution.

"What the customer really needs", is a very difficult thing to nail down. The extent to which we burden information technology solutions with the qualities of flexibility, low maintenance, user-friendliness, robustness, etcetera; we are adding overhead to the process as perceived by our client. If we take the approach that we will burden the solution only to the extent that is necessary to satisfy the client, we may deny the business some level of security. For example, we can deliver systems with or without backup capability. The client might demand that the cost of such be avoided even when the business need clearly demands that the feature be provided. This is where client expectations management can benefit from establishing a "Service Level Agreement."

Service Level Agreements establish a clear statement of agreement between the service provider and the client of what features and functions will be delivered and with what level of service. The service levels might include response time, failure recovery time, history keeping and any other points that can be measured and that appear to either the client or the service provider to be critical to the success of the solution. Service Level Agreements imply performance metrics and measurements. This combination provides another of the key concepts of the architectural approach. That is, Performance Standards, Metrics and Measurements should be included in all designs.

The final concept is that of "Life Cycle Management." By this I mean that we should look at every process that is employed in the delivery of information technology solutions and manage it as a life cycle that begins with the initial idea and extends through its useful life and into retirement or replacement. By considering the Life-Cost and the Life-Support and the Replacement-Cost of each solution, we establish a framework for business level decisions about them. It is very easy to spend money on the Features and Benefits of a project but to spend money on the Support and Replacement often leads to great disappointment.

What has been laid out above is an evolutionary view of the process that I went through as I learned a set of principals for the successful delivery and support of several information technology solutions to business problems and opportunities. Without too much insight and analysis, one can also see that the Total Quality Management philosophy has also been stated. This is a very important link that allows one to create a unified set of operating principals that span the entire information technology organization and allow it to fit into the business methods and practices using a compatible set of goals, measures and language.

Establishing the basis for Total Quality Management to be Continuous Process Improvement to ensure Total Customer Satisfaction we can then establish a model for Information Technology Deployment to be a set of processes that work together to provide solutions to our client's business problems. What comes next is to develop processes that help us to identify and resolve business problems. Some of those solutions will employ Information Technology in the form of computers, networks, telephones and other hardware tools. Some will employ software tools both home-grown and purchased. Some will merely be resolved by the process of "Process Improvement."

What is the Architecture?

Information Technology Architecture provides a framework of reference for the construction of standard practices and methods for the delivery of information systems solutions to business opportunities. Architecture is positioned as a reference point not as a standard or policy. It should be viewed as a set of known practices and solutions to previously resolved issues and therefore it provides one way of doing things. It is important to keep Architecture in its place. That is, as a reference not as a bible or law. In as much as an Architecture addresses the form, fit and function of a set of previously detailed situations, it can not be relied upon to cover all cases or even to provide the best approach to an old problem. The domain of Information Technology Solutions and Tools changes too rapidly for that.

In the following paragraphs, many of the aspects of the Architectural Approach will be presented. The approach, as I refer to it, is made up of a set of processes and models that hang together well throughout the life cycle of the Information Technology Delivery Process.

The Information Technology Life Cycle

Conceptual	Tee Shirts and a logo. Work by people with an idea.
R&D	Weekly Industry News ads and marketing activity.
Emerging	Insiders see alpha-ware, prototypes, development of support chains.
Preferred	Training complete and production systems under construction.
Current	Most actively chosen and used technology.
Declining	A new preferred tool has arrived.
Retired	Design-out efforts under way. Maintenance remains.
Bothersome	Cost exceeds value.

The Conceptual phase refers to that period of time when the technology is evolving from an unorganized set of goals, potential applications and just plain neat technical stuff into a work product that will sustain itself across a sufficiently large number of applications. Often such techno-soup never yields a true technology. It is unwise to include conceptual technologies in an architecture. However, it is very important to analyze the potential of such. Hewlett Packard committed enormous resources to the Precision Architecture even in its Conceptual Phase because of the apparently high potential.

The R&D Phase can manifest itself in a number of ways. If it is an industry based technology, there will be as much customer based research as there is product based. These technologies can often be found in the industry watcher's newsletters and in market research focus groups. In this phase, the products still have only a probability of ever existing and generally, the future look and function will not closely resemble the early prototypes.

Look for the essence of the product not the exact form in this phase. Avoid including the exact technology but do consider using the underlying themes of such in an architecture. The principal value of these technologies is that they are leading indicators of market shifts if not paradigm shifts. HP New Wave was not a barn burner of a product however it was a turning point for human interface just as Xerox Star begot the intuitive Graphical User Interface.

The Emerging phase is heralded by appearances at trade shows and what appear to be sales advertisements in the trade journals. Actually the technologies are more often in the vapor-ware domain while beta testing and distribution evaluations take place. The vendor is the key to architectural element decisions. Those that have a good track record can often be counted on to deliver. It is a good idea to establish good relationships with a small number of key vendors so that inside information about the products can be accessed. It is often a necessity to include emerging technologies in the architecture when bleeding (sic) edge technical solutions appear to be worth the risk. Companies like Bose grow on innovation. We find that we must push the envelope of our fundamental technologies at all times to remain at the top of our markets.

Preferred technologies have arrived. Not necessarily in a global sense but rather to the degree that a clear and demonstrable line of support from marketing and sales to training and consulting and through the delivery process has been established. These technologies form the tool sets of choice although they will often be employed only on the most recently commenced projects. During this phase, the set of weaknesses and strengths should be documented and guidelines for application should be drawn up in detail. It is also during this phase that the relationship with suppliers of the technologies should be nurtured and a certain degree on influence over the configuration should be obtained. This is often best accomplished through user groups but "significant account" status and beta testing participation are alternative routes. Progressive vendors like HP, Apple Computer, Cognos and O'Pin Systems have demonstrated a strong desire to keep their products in this phase making them responsive to customer desires and thereby perpetuating their markets.

Current technologies are those that have become pervasive like the light switch. Everyone knows how to use it and fully understands what it does. It is important that this phase of the life cycle be sufficiently long to obtain financial value from the investment made to obtain and implement it. Another important factor to consider is to make sure that this phase is not prolonged past its useful life. This is a delicate balance that separates the technicians from the accountants in a hurry because we generally get more than the accounting asset life out of the technologies. Software lives about two to four years in a growing company. Personal computers are obsolete within two years. Mechanical equipment will generally wear out before a five year life and we often outgrow facilities before their ten year life is up. Before a technology is acquired, the pay back period should be established and considered. Planning for the replacement, enhancement and upgrade of capital assets should begin before they are put into service.

The Declining phase begins when a replacement technology has been selected and its Preferred phase is under way or when the necessity for the technology is eliminated. Just as a plan for the acquisition and training ensures a smooth transition into use, a phase-out plan should be created to ensure a smooth removal from service.

The Retired phase begins when all new development with the technology is suspended and a design-out process has begun. The retirement process addresses the urgency of removal from service as well as the program that will result in its full removal. This is generally the most difficult phase to get through because the old "if ain't broke, don't fix it" attitude prevents us from getting the work effort under way. Avoid being put into the position of a required removal from service as might happen with computer operating system changes or telephone equipment upgrades. It is difficult to explain that you need a new computer to perform call accounting just because you changed some leased lines.

The Bothersome phase for a technology begins when the cost of employing it exceeds the value gained from it. This happens rather quickly for mechanical devices like disk drives and printers as their maintenance costs accelerate out of sight but the more subtle burden of a retired 3-G or 4-GL produces a prolonged burden of maintenance cost and staff training. In the worst case, it may create a roadblock to progress in other areas such as CPU upgrades that would improve performance and reduce operating cost.

By dealing with technologies as resources that are implemented through hardware and software devices they in turn should be managed as capital assets. Such capital assets have an initial cost and value. The value increases for a while as we learn to exploit it then it levels off and finally it declines. Along the same time frame its cost will change. Technologies are elements of a life cycle process, each phase presents another process that should be optimized and continuously improved. Improvement in this context is a value analysis.

The Characteristics of Information Technologies

Each technology should be:

.. Flexible so it can be applied to multiple and diverse applications and so it can interconnect with other technologies.

.. Extensible so its capacity can be scaled to a wide range of applications.

.. Cost-Effective so it will fit within the financial support structure of the client and the business application. (The full cost of ownership equation must be considered over the full life cycle including cost to extend its capacity, environmental requirements, maintenance and operating costs)

.. Supportable without the necessity for additional or expert support staff and so there is a minimal requirement for building unique support channels.

.. Manageable so operating performance characteristics can be measured and its capacity can be controlled.

Technologies are moving targets. The hard call is "When to get on and When to get off." The architectural approach deals with each technology in both time and application domains where the application drives the decision. High risk technologies might be ok for rifle-shot applications whereas we would want only mature and stable ones for general application. This is where the skill of the technician comes into play.

Quality in Information Technology

You can have it Cheap. you can have it Fast. You can have it Right. .... Pick Two. This is the application development triangle. Either explicitly or implicitly, we give this choice to the client who will invariable pick ASAP and Low Cost. What the architectural approach attempt to do is to use models, processes and technologies to create a zone of compromise that makes it easier to achieve Quality without trading off Expense and Expedience. 4-GLs are good models for this. They allow us to build better and more functional systems both faster and with higher quality.

To implement Total Quality Management in Information Technology, a process orientation is particularly well suited. This is opposed to a more conventional goal or results orientation which focuses on end products rather than methods and practices. Results oriented management tends to favor short term goals, individual effort, hard work and haphazard methods whereas a process orientation promotes long range planning, teamwork, smart work and uniform methods.

The management structure that supports a results orientation must be hierarchical and militaristic in nature. This demands that, to a great extent the management specifies Who, What, Why, When and How for all work products. This ensures that the manager who is being held responsible for the results can control his/her destiny explicitly. A process orientation is organized as follows.

Employ the creativity and skill of people to with the power of technology to achieve more effective ways of performing business processes.

We then...

Organize a set of processes to support the business functions.

Organize competencies to support the processes.

Organize tasks to support the processes.

Organize people to support the tasks.

Organize management to support the people.

Apply Total Quality Management Concepts across the process life cycle.

Organizing around processes and positioning management's role as a facilitator and resource manager implies a flexible organization of the staff that does the "real work." Project teams, duty rotations, experts, consultants and out-sourcing all fit well into this scheme. The idea is to organize a set of process steps (work products) that can be delivered by the staff with the information technology tools to produce business solutions. What is critical is that all parties involved in the delivery of a business solution work as a team with clearly defined and shared goals and responsibilities.

Management may have to take on a new role as a transition is made to Process Management. The key difference is how issues are handled and resolved. The manager's role should be to deal with the process weaknesses that create the issues, not the issues themselves. If, for example, a project is late; a results oriented manager would jump in to find out how to get the project done as soon as possible and if time permits (which is rarely does) will later come back to see why the issue arose in the first place. A process oriented manager would review the project progress to date; isolate the processes that contribute to the delays; identify the process weaknesses; then mission the project team to compensate for or resolve those weaknesses. This can be a rather large shift in style which demands that discipline and trust replace brute force.

The competencies that support the processes include the people, tools and machines that are employed to deliver a business solution. Competencies which are part of the architecture will generally be acquired from internal sources. Those that are not part of the current architecture will generally be out-sourced or developed as part of the process. For example: We need a center of competency for operating HP3000 mini-computers if we are to successfully deploy a business system that demands the use of one. At the same time, we might connect to a service bureau for EDI services. We might choose to out source the EDI interface and processing and build only a minimal support service for the EDI application internally.

Out sourcing has become popular recently. Among the reasons that companies offer for out sourcing are: cost reduction and strategic opportunity. Of the two, the one that appears to be valuable in the long term is Strategic Opportunity. If a third party can provide a service at a lower cost than the in house group does, it is generally true that there was waste in the internal operation. There are companies that provide a service to measure internal waste by bench marking them against similar operations. My experience from these companies is that the out sourcing company earns its profit from their clients former waste that they are able to eliminate.

This leads to the second reason for out sourcing. By employing an outside operation, the political, organizational and other types of baggage that prevents internal right-sizing can be avoided. There is therefore, a tactical advantage to letting someone else implement your strategic change of direction. Specifically, allow the out sourcing process implement the right sizing strategy for you. Other strategic opportunities for out sourcing include short term staff increases, particular expertise that is only needed for a short term. Thereby, turning over processes to an outside staff while the internal group builds for the future. To me, the most appropriate guideline for out sourcing is to buy what you can afford, or choose, to do without and to own what is critical to your company.

When we organize people around tasks that might change with process development and operation demands, we introduce a requirement that breadth is equally important to depth of knowledge. Indeed, jobs do have a tendency to change quite regularly in this organization. To maintain consistency, key positions of expertise are maintained while others may swing wildly from one work focus to another. This is a difficult issue to resolve with traditional Human Resource Management programs and especially with traditional Personnel Evaluation Processes. I have seen no successful ways of dealing with this conflict up to this point. I hope that more can be offered on this in a later work.

The critical characteristics of Total Quality Management as they apply to a Process Management style are:

Goals must be customer driven.

Customer's perception defines quality.

Leadership must be strong to manage apparent conflicts.

Continuous improvement of all processes leads to lower cost of quality.

Plans are based on ideas and concepts but actions are based on facts, data and analysis.

All parties in a team must participate in the process.

We are in existence to support our clients who in turn exist to support theirs and so on up the chain to the consumer that pays our salary. The Total Quality Management orientation turns the measurement of our success over to our clients directly. It is their perception of our ability to meet their needs that establishes our quality level of service.

This can present a paradox "want versus need" as was stated earlier. Strong leadership and "expectation management" are needed to resolve, pro-actively, the inclination of a client to go for Now and Cheap than later hold the lack of Quality against the provider. Strong leadership is also required to help the staff accept and flourish in what will often appear to be a department without clear rules of order. And most importantly, strong leadership is required to arbitrate with clients and other departments who are not practicing a Total Quality Management approach.

It has been my experience that a customer dedicated person is much less tolerant of poor service than one who is not so dedicated. Also, a person who is dedicated to continuous improvements is not tolerant of continuous impediment from others who do recognize the need to improve their services. Although Total Quality Management works best when all members of the team (the entire company and all clients) practice it together, it falls upon management's ability to lead to resolve these apparent (and therefore real) conflicts.

Total Quality Management of Processes

The first order of measure for a work group is the delivery of work-products each of which is the result of a work process. Total Quality Management assumes that by continuously improving the work processes, the quality of the work products will improve. For those who might feel that cost is being ignored; Total Quality Management also assumes that process improvements reduce cost not only by reducing waste and error correction but also by process simplifications that result from studying the processes during the improvement cycles.

Because work processes and people are not perfect, a certain amount of variation will result in the stream of work products. When this variation wanders beyond a certain margin of acceptance, the client notes a defect. Quality

Management is therefore the management of variance and the margin of acceptance. Both are fair game in our efforts to succeed. We can manage variance of work products if:

Processes are defined for each activity.

One individual is responsible for each process.

Defects are attributed to process variations.

The entire organization accepts the challenge to reduce variations.

The entire organization accepts the challenge to specify the work products of each process.

To this point some might think that this only applies to some processes or that it can't apply to some others. The method applies to all aspects of Information Technology delivery including: Personnel Selection, Training, Motivation, Budget Management, Quality Assurance, Software Development and Customer Account Management. The trick is to define the process flow, establish the expected service levels and develop the competencies necessary to support the combination.

As a slight digression, I introduce the real-life condition that there are two types of processes; convergent processes and divergent processes. At our plant in Canada, we manufacture wooden cabinets for loudspeakers. As we repeatedly cut large panels of wood into small panels of wood, the thickness of the cut decreases and the quality of the cut deteriorates. This is a divergent process. We begin by making cuts as wide as is acceptable and allow the blade condition to deteriorate until the cut approaches the minimum width. We then replace the blade and we start over again. This is not divergent because it moves away from a norm. In fact it moves towards the norm at first and then away from it. It is divergent because it never ends. Marriage, Child Raising, Income, and Automobile Repairs are all divergent processes. That is, the effort never ends.

Travelling from Boston to Atlanta is a convergent process. At least I hope that it will be for me. I will leave Boston and I will eventually arrive in Atlanta. If I must repeat the process I might improve my route, schedule or cost as a set of divergent processes until I can get there instantaneously for free.

It is best to consider all processes as divergent and at the same time identify when to stop improving one process and move on to another. A good method is to establish improvement goals and improvement evaluation periods. Your barber or hair dresser is a good example of one whose success always depends on knowing when close enough occurs.

To achieve the goal of having manageable processes we can take the following as a set of premise:

Repeatable processes can be counted upon to yield consistent results.

Measurable processes can be monitored for performance to standards.

Controllable processes can be tuned to reduce variations.

Scaleable processes can support various sized work loads.

What remains is to provide a mechanism to determine when to scale up or down and when to adjust for variances. The electronic analogy is to create a feedback loop which compares the desired output to the actual output. In an electronic amplifier for example we compare the shape and magnitude of the output signal back to the input signal then send the difference into the amplifier to correct the process. That works ok when the output should be some known function of the input.

In processes (systems) that involve people, the desired output changes not only as a function of the input but by a myriad of other factors. "Yes, I know that I said I would only run this at month end but....;" "How was I to know that some customers would want to pay more than once in a day;" "By backups, I thought you meant a second computer not just tapes." To reconcile the differences, customer satisfaction must be evaluated frequently enough to avoid the effects of business change. Quarterly at a minimum but Monthly or more frequently if possible. Bose, Technical Services is set up to request a quality assessment for each work product as it is delivered.

The closed loop design for Information Technology must establish the customer at the end of the process to evaluate and feed back quality assessments. It must also establish capacity monitoring and management for all resources including machines and people.

This prompts another digression. Time, Money, Machines, Tools and People are the principal resources that the Information Technology Manager has to work with. Treat them all with the same respect and attention. They are closely interrelated, they have similar limits and they can be traded off among themselves. The resource that makes it all work is the people. However, not only is their effectiveness and efficiency limited by their access to tools and machines but, their motivation is also effected by the extent to which their effort (not skill) is used to compensate for lack of other resources. Do develop resource management plans for human resources. Do also learn how these various resources can be managed to sustain the business needs not only of the customers but also of the Information Technology Organization.

Architectural Planning

Up to this point I have introduced a set of models and concepts that describe a process for relating the activities of Information Technology developers to an architectural blueprint, to their customers, to the processes that they employ and to their management. Now, let's link it to the business.

The title of this paper is actually opposed to the methods that would be ideal. We would like very much to have a comprehensive business plan onto which we could map a set of Information Technology solutions. However, in the real world we find that the "Current Phase" of a technology is so long that we have to fit solutions to available tools more often than the other way around. Let me first describe the ideal.

Business Process Analysis

To improves a business process:

It is first decomposed into a set of sub-processes each of which is described in some detail.

The result of the process is clearly defined and a set of quality metrics and acceptable variations for the work product is established.

The inputs to the process are detailed and their known variations are assessed to determine their effect on the variation of the work product.

Variations of the inputs are minimized.

The process is simplified to the extent possible without automation.

Automation is employed to further simplify or reduce variation.

Note that automation is the last step in the improvement process. Automation is employed to simplify or reduce variation within a process. This keeps the focus on the inputs to the process and avoids automating the problem by simply making the process faster.

If we had before us a comprehensive list of process steps that required automation, we could survey them and choose from available and emerging technologies a set of tools and machines that would best fit the overall project of process automation. The goal of the Information Technology Architecture is to provide a set of solutions that can be applied to the spectrum of business process automation tasks.

This would be great for the moment. When the next process automation effort came along, it may or may not lend itself to resolution using the then current tool set. We would however be obliged to try to use what we have going rather than introduce a new tool.

I refer to this as "ploughing around rocks in the field." Some of these rocks are big ones like MPE-V or MPE/iX or Apple Macintosh or TurboImage. In fact, some of these rocks are indeed "best of class" solutions for their initially intended purpose but they just don't work well together or even don't fit the new problem. Simply stated, that is the way it will be. As we introduce new technologies, we introduce new rocks to plough around. If we simply continue to add and do not remove, we will overextend the ability of the staff to support all of the tools and we will overburden our ability to pay for the external costs of maintenance and upgrades.

To begin the architecture development, we can document the current set of tools and machines and establish their locations or desired locations on the Information Technology Life Cycle. Once we have this chronograph, we can add new capabilities that might be desired with the potential technologies that will provide the desired services. We then develop a program for evaluation and selection of new technologies and for the retirement of unwanted technologies. Sounds simple and it would be if there were not so many stake holders in the current tool set and so many opinion holders in the targeting of available or emerging technologies.

An architecture should hang together. For example, If client-server technology is desired and the current business systems are deployed on HP3000 MPE/iX systems with a current population of Apple Macintosh personal computers (two best of class choices) the architecture should show Apple Macintosh Clients and HP3000 Servers. That technology tool set does not exist and we therefore need a work around. On the other hand, if a combination of HP3000 MPE/iX and HP9000 HP/UX systems are included in the architecture, then Cognos 4-GL and HP Allbase or Cognos Starbase looks good on the architecture not only because they carry the same names but because they actually are the same architectural elements.

Getting back to the Client/Server dilemma. Suppose we could connect Apple Macintosh to Cognos StarBase on HP/UX and from there get to the HP3000 MPE/iX system data. This would imply that the data must be in some data structure that can be bridged from StarBase on HP/UX to StarBase on MEP/iX and from there to a data structure such as TurboImage or ORACLE. We can also take the architectural approach of migrating data from all existing structures to StarBase so that the connection can be accomplished with less overhead and complexity. Whatever choice is made, an implementation process must be defined and executed.

The Mapping Process

A business process requires ad hoc reporting from a rather large data base. The straightforward approach is to let them have at it with Cognos Quiz and satisfy the apparent request "off the shelf." From experience, we learn that this is not generally a good practice when the data base is also used for on line transaction processing on a machine with limited capacity. Another approach would be to convince the customer to accept overnight reports thereby starting off with a less than optimum solution. The process decomposition and architectural design approach might reveal that neither approach could work.

Ad hoc reporting is often an iterative process where a report or a set of reports is produced over some period of time. If the on line transaction processing system data bases are used, each iteration of the report may yield different results confusing the requestor(s). What the client needs is a stable reporting base. We can create this by copying the production data base to a reporting data base. At the same time, we are managing the capacity of the computers and find that the OLTP machine needs an upgrade soon. We know that hardware upgrades imply software upgrades to many tools. What if we add a smaller machine to handle the batch processing and ad hoc reporting thereby reducing the load on the OLTP machine and obviating the necessity of upgrading many software tools that are not really providing any more service at the upgraded higher price.

By resolving the ad hoc reporting request down to a set of input, process, processor, and output requirements, the potential for providing a more generic architectural solution presents itself. The architectural approach that results is to separate batch reporting from update processing across two machines. Since the batch reporting is done at night and the ad hoc reporting during the day, the second machine capacity can be less than otherwise might be required. By reducing the load on the OLTP system, its hours of service are increased, its performance life is increased and its cost of ownership remains the same.

Now back to Mapping the Business to the Technology. This is called a technology push. It rarely works. Avoid it if you can but if you must make a forced fit, be sure to include more appropriate recommendations with the one that is available. Reflect the difference in the cost of ownership equation and in the service level agreements. Also log this less than optimum solution with others and their costs for use in your next efforts to obtain the resources for a proper solution.

Summary

The mapping between Business and Information Technology is a two-way affair. The degree to which the two map into each other appropriately determines the probability of success for the information technology delivery group. Using an architectural blueprint of business processes and technical facilities enhances the ability of the information technology staff to deliver quality solutions expeditiously at low cost. Process Management employing Total Quality Management methods helps to ensure the link between the client and the Information Technology solution provider. A Life Cycle approach to resource management helps to ensure a progressive work environment. Most important is organizing around people who support processes. This gives management a facilitation focus which fosters the empowerment of both the staff and the clients. My Architectural Approach is a combination of all of these concepts and methods.

Paper # 6003

Meeting the Network Challenge: Connecting your PCs, Hosts, and LAN Services

Evan Kaplan
Walker Richer & Quinn Inc.
2815 Eastlake Avenue East
Seattle, Washington 98102
206-324-0350

The Problem

The migration from the asynch world to the network domain is now in full swing. The percentage of devices connected via network links has risen steadily over the last few years and promises to continue to grow exponentially over the next decade. In a "textbook case" the transition to networked computing would be driven by the obvious benefits of the new medium, i.e., faster data transmission, dramatically increased transmission reliability, and simpler and relatively inexpensive wiring schemes. Yet while these are all critical, the real driving force is something quite different.

The most influential single factor in the dramatic shift towards network connectivity is the rise in popularity of PC LANs—Novell, Banyan, and LAN Manager (LAN Man) network operating systems (NOS). The virtual disk capability at the core of these NOSs—with its file sharing and storage benefits—appears to be something that corporate end users cannot live without. The movement towards LAN-based E-mail and the efforts the NOS vendors are making in the areas of network management and host-server communication are likely to intensify this trend. All of these factors point to an increasingly important role for the LAN server in the corporate computing environment.

The PC LAN has normally been implemented at the workgroup level, where some basic infrastructure must be put in place. PC LANs require Ethernet, Token-Ring or Arcnet connections at the desktop. Network cabling is brought in, and network interface cards installed on the PC. NOS's and their respective servers are installed and configured.

Although PCs have gained LAN capabilities, they usually have maintained serial connections to host computers, too. The typical shop has continued to rely on an a variety of these connections—direct connections to a host port, or through a dedicated terminal controller, data switch, or, in some cases, a MUX. But even with all of these connections, the user still has difficulty accessing more than one host and even greater difficulty accessing more than one type of host. (For instance, many users at an HP site may need access to an HP 3000 and an HP 9000.) In addition, just maintaining and supporting these connections is a job unto itself.

The result of these two separate activities is connectivity redundancy. A PC that is on a LAN and has serial connections to a host has a serial and a network drop. The network manager still must handle two separate configurations.

So the question becomes: What is the network manager to do to handle communication with these disparate services and their respective connectivity schemes?

The Options

The ideal situation is to provide one wire from the desktop to give users access to the variety of hosts and PC LAN servers on their local and non-local networks. In order to achieve this ideal, all of the hosts and servers must have either a common language or a translating device. Getting beyond the "language barrier" can be achieved in a variety of ways. The MIS environment can be designed so that all the host and servers use a standard protocol suite such as TCP/IP (transmission control protocol/internet protocol). Another option is to build gateways or protocol converters between the various devices to allow for "language" translation. A third option is to give the desktop device the multilingual capability to speak the languages it needs to speak when addressing each of the respective nodes. Each of these approaches has its relative strengths and weaknesses, which I will discuss in this paper. But first I should give an overview of the primary languages we are likely to see in an various MIS environments.

Protocols: The Hosts

Hosts: HP 3000

Only two protocols will establish a network terminal session with an HP 3000 host. One is AFCP(Avesta Flow Control Protocol). AFCP is the protocol that the DTC uses to communicate directly with the 3000 host. AFCP is not available to the desktop. It is an

HP proprietary protocol used exclusively for communication between HP 3000 XL machines and the DTC. Serial inputs or, more recently, Telnet inputs to the DTC are converted to AFCP for communication with the 3000. Telnet is the standard virtual terminal protocol for the TCP/IP transport.

If you want communicate directly from the desktop to the 3000 over a network link there is only one option—NS/VT. NS/VT stands for Network Services Virtual Terminal. NS/VT is the equivalent to Telnet, in that it is designed to support virtual terminal communication over the TCP/IP transport. But NS/VT is proprietary to the HP 3000 environment, while the Telnet protocol is supported on virtually every host platform. (One third party vendor offers a Telnet implementation running on the 3000, but this implementation supports only the "classic" machines.)

Hosts: Unix Hosts, Including the HP 9000, and Other Hosts Supporting TCP/IP

The standard means of creating a terminal session with a UNIX host is through the Telnet protocol. For all UNIX systems (including SUN, DEC ULTRIX, RS 6000, etc.) the TCP/IP protocol suite is standard. TCP/IP includes Telnet, FTP (file transfer protocol) for file transfer and NFS (network file service) for file sharing. All of these protocols are application protocols that run on top of the TCP/IP transport protocol.

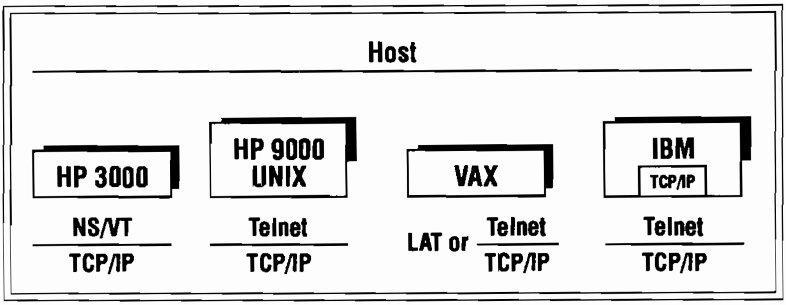
TCP/IP is gaining rapid acceptance as the internetworking protocol. Almost all of the major minicomputer and mainframe manufacturers are supporting a TCP/IP implementation. Most noticeable among these is IBM. IBM recently released a barrage of products for the mainframe environment designed solely to enhance TCP/IP connectivity. A standard equivalent to Telnet has been defined for the 3270 datastream (IBM's datastream for mainframe to terminal communication) to support virtual terminal over the TCP/IP transport. The standard is known as TN3270 (Telnet 3270). A desktop device with an implementation of the TN3270 standard can communicate directly with an IBM mainframe equipped with TCP/IP, or alternatively a TCP/IP-to-SNA gateway device.

Hosts: Digital VAXs

It is not uncommon to see stand-alone VAXs or VAX clusters in an HP environment. A fairly large percentage of HP shops need VAX connectivity. This trend is definitely on the upswing as each manufacturer touts the interoperability of its respective platform. To communicate with a VAX from the desktop, you have two choices. The first is LAT

(Local Area Transport). LAT is a DEC proprietary protocol that supports virtual terminal communication between a DEC host and a desktop device or terminal server. LAT communication capability is bundled with all VAX hosts. LAT is an efficient protocol for local communication, but because it is not routeable it is not used in remote communication.

The other alternative for the VAX is TCP/IP. TCP/IP is purchased as an option for the VAX. It is available from DEC as well as a variety of third party vendors. TCP/IP on the VAX is very common today and its popularity is definitely increasing because of DEC's failure to deliver a true OSI Implementation. With TCP/IP on the VAX, the user can Telnet to a VAX host from his/her desktop.



Protocols: The Network Operating Systems

Communication with the host systems is only half of the picture. The other half is communication with the network operating systems.

Novell

To communicate with the Novell server, the desktop user needs to run the NetWare client on top of the SPX/IPX (sequenced packet exchange/internet packet exchange) transport protocol. Currently the SPX/IPX transport is the only method that will create a session with a NetWare server.

LAN Manager

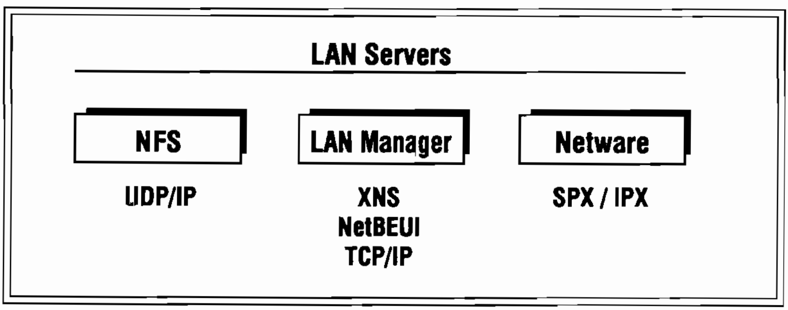
LAN Manager in its various flavors (Microsoft's, HP's, AT&T's , Pathworks, Ungermann Bass's) is transport-independent. Because LAN Man depends on a NetBIOS transport layer interface, any transport protocol that can support the NetBIOS interface can be used for LAN Manager server communication. Some of the more common are NetBEUI, shipped with Microsoft's and IBM's versions; NBP, shipped with 3 Com's implementation; and XNS, originally shipped with Ungermann Bass's LAN Manager. These protocols are efficient for local area communication but lack capability for wide area communication. Recently a number of companies have begun offering TCP/IP as an optional transport for LAN Manager, primarily because of its wide area and internetworking capabilities.

NFS

While NFS is not officially a network operating system, in practice the NFS standard is commonly used for simple file sharing and often in place of the the "name brand" NOSs. All NFS implementations require the UDP/IP transport (a subset of the TCP/IP transport). NFS servers are available on many platforms but are most commonly used with Unix-based hardware. The most popular implementation of NFS is PC-NFS from Sun Microsystems.

Banyan

Banyan uses a combination of its own exclusive protocol and the TCP/IP standards.



Simplifying the Milieu

Yet outlining the situation is only the first step. Given an understanding of the problem and a good feel for the "linguistics", how does a systems manager go further to enable the communication?

Let's take a look at the three basic strategies:

Homogenize Your Network Communication.

One strategy that many corporations are pursuing is standardizing the language they use for all their network communication. Originally the plan was to adopt the OSI protocol suite, but the lack of significant commercial OSI products has accounted for a very slow adoption of these standards. Instead of OSI, many corporations are standardizing on the TCP/IP protocol suite. Its primary advantages are its widespread availability and its track record.

There are benefits and drawbacks to the single protocol approach. The benefits are that a single protocol eases network management. The routers, the network management utilities, and even the network engineers have to be concerned with only one type of protocol. Productivity is increased by smoothing learning curves, and even the cost of equipment may fall if the chosen protocol is widely supported.

One drawback to the single protocol strategy is that it is difficult to optimize a single protocol for the variety of communication required on a corporate network. For instance, while TCP/IP is a very reliable transport for a variety of communication links, IPX and NetBEUI are optimized for LAN communication. By standardizing on TCP/IP, the network manager may be sacrificing some performance. Another drawback is that despite wide support for a standard protocol, there may be popular (perhaps even required) applications that still do not support the standard. Novell NetWare's current lack of support for the TCP/IP transport is one example of this. Finally, while today we see most of the significant players in the computing world espousing an "opens systems" strategy, it difficult to believe that the struggles for market dominance based on proprietary architectures is completely over. If this is true, the network manger may be exposed to some risk by committing all of his/her networking resources down a single path.

The benefits and the drawbacks of the single protocol approach mirror the ongoing struggle between standard and proprietary approaches to network architecture.

Use Gateways or Terminal Servers to Translate Your Protocols.

Many network managers have used gateways or terminal servers to minimize the linguistic difficulties. Gateways essentially translate specific protocols between disparate senders and receivers. This is done in a variety of ways. The gateway can encapsulate one protocol within another. The Novell server performs this function when it encapsulates IPX packets within TCP/IP datagrams in order to support wide area communication. The NS LAN gateway from HP does the same with IPX and NS/VT.

Some terminal servers can perform essentially the same function when they act as a session management device. For instance, in the DEC world it is common for a LAT terminal server to advertise a Telnet service (host). The user might use LAT to log in to the terminal server and use the Telnet capability within the server to establish a session to a UNIX host. In the HP world similar functionality is available with the new Telnet Access and Express products, which allow the user to connect to the DTC via Telnet. These inputs are translated to AFCP for communication with the host. Since the DTC requires one of the connections to be serial, it cannot support this kind of connection from a dumb terminal.

The terminal server's primary advantage is that it can support dumb terminals as well as PC connections. The gateway's primary advantage is that in low traffic installations it can be used for functions other than protocol translation.

The drawbacks to protocol translation is that it is simply a lot slower than direct connections. The stripping and encapsulating process in the gateway and the necessity for dual session maintenance in the terminal server tends to decrease performance. In addition, these devices have a tendency to become bottlenecks in the network and **create** simply one more hardware node to manage.

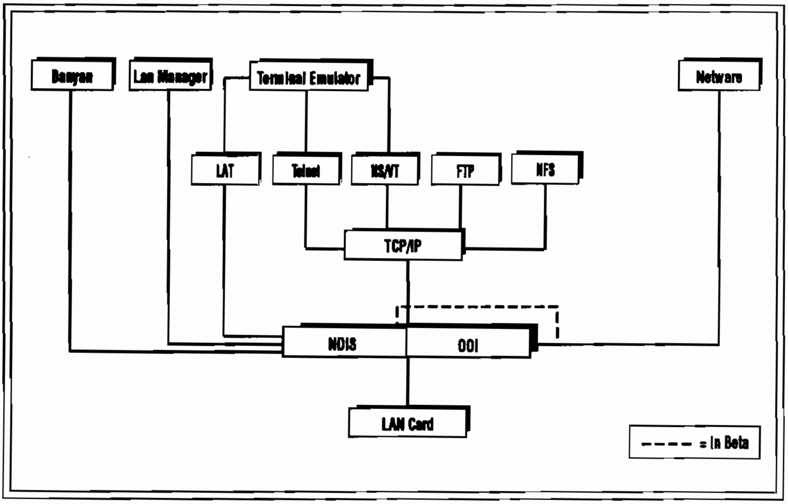
Make Your Desktop PC Multilingual.

The final strategy is to give your PC the ability to make connections using a variety of protocols. In the past this was virtually impossible. Network card manufacturers provided only direct drivers for their adapters, and protocol providers wrote their code to interface to these drivers. This meant that a single network card could handle only one protocol at a time. In order to communicate using another protocol, the PC user would have to reboot or install an additional network card. Recently, two standards have emerged to allow a single network card to handle multiple protocol stacks. These standards are NDIS (network driver interface specification), sponsored by Microsoft and 3Com, and ODI (open data-link interface), sponsored by Novell. The two standards have changed the entire scope of PC communication. Network card manufacturers now supply NDIS and ODI drivers, and the protocol providers write their stacks to comply with these drivers. Now the PC can support multiple protocols with a single network adapter.

A large number of companies today support PC-based TCP/IP stacks that are compliant with the NDIS and ODI standards. Some support LAT and TCP/IP. Only two support the NS/VT protocol at all. In addition almost all of the network operating systems have designed their transports to run over NDIS (including Novell, according to a recent announcement). This means that PCs equipped with the compliant protocols can support multiple Telnet, LAT, and NS/VT sessions while maintaining their LAN server connections.

There are significant advantages to the multiprotocol PC. The most obvious one is now the PC can communicate with a variety of different hosts and servers without being concerned about which protocol the node is using. This is especially important in larger corporations or universities where users often have to communicate with a variety of different hosts and servers. The communication itself tends to be dramatically faster, because it is limited only by the particular network's bandwidth and routing capability. Another advantage is that this kind of solution allows network designers flexibility in approaching the whole protocol dilemma. They do not have to buy expensive host upgrades or gateways. Nor do they have to migrate to a new standard immediately. Network designers can phase in the "open systems" standards while maintaining those devices that require the proprietary protocols, allowing use of both old and new technologies concurrently.

Of course, there is some price to pay for this kind of functionality. The biggest one is the PC memory overhead. With all of these protocols loaded, the user will consume a significant portion of conventional DOS memory. This is mitigated by the fact that many of the protocol providers are beginning to supply Windows-compliant protocol stacks that can be installed in extended memory. In addition, many of these suppliers allow the protocols to be unloaded from memory on demand or loaded high with memory managers or DOS 5.0.



Summary

The objective is clear. From the network engineer's perspective, one wire from the desktop should allow the PC user access to a variety of host and LAN services. This paper has defined the communication problem in the context of the proprietary and "open systems" architectures that define today's networks, and the barriers they present to the potential integrators. It has also laid out some of the options the network engineer or MIS manager may consider in solving these thorny problems.

It is my position that the key factor in deciding which strategy to pursue is flexibility. Flexibility is important because the transition towards standards seems to move in waves, and it is difficult to know whether we are just gathering power or cresting. HP'S Network Services, DECnet, and IBM's SNA are still the dominant players within their respective worlds. TCP/IP is showing tremendous promise in terms of availability and applicability, but it is not a panacea. OSI is looming as a potential replacement for TCP/IP. And the NOS vendors seem to be pursuing a variety of strategies. It is difficult to predict the network architecture of the future. The safest, surest course for the present is to position your users to respond to whatever changes the future brings.

THE MULTIPLE "C'S" OF MAKING A SYSTEM CHANGE

Mark H. DeVries
Blodgett Memorial Medical Center
1840 Wealthy SE
Grand Rapids, MI 49506
(616) 774-7513

WHO WE ARE

Blodgett Memorial Medical Center (BMMC) is a non-profit community/teaching hospital, founded in the late 1800's, licensed at 410 beds and currently operating at 343 beds. BMMC is a "Center of Excellence" for the Cardiology, Neuroscience and Orthopedic specialty areas.

The staff of the Information Systems Department consists of:

12.5 FTE's including:

- (1) Director of Information Systems
- (1) Operations Supervisor/System Manager
- (1) Network Manager
- (1) Clinical Coordinator/Analyst
- (1) Financial Coordinator/Analyst
- (1) Ancillary Coordinator/Analyst
- (1) Senior Analyst/Quiz Programmer
- (1) Administrative Secretary
- (1) Staff Programmer
- (3.5) Operators

In an effort to improve the quality of service rendered to its patients and physicians, BMMC began in 1988 the process of evaluating various alternatives for a replacement to the current hospital information system (HIS).

The automated environment under which the hospital was operating in 1988 consisted of the following:

- **MEDPRO**
  - Applications:
    Order Communication
    Pharmacy Management
    Census Registration
    Order Prompts
  - Hardware: Four-Phase
  - Software: HBO & Company
    Proprietary Operating System

- **IMPACT**

    . Applications: All Financials including Case Mix

    . Hardware: HP 3000 Series 58

    . Software: Cobol and Powerhouse® mix
    In-house supported to extent possible

- **CLINSTAR**

    . Applications: Radiology Management

    . Hardware: Data General MV400 Eclipse

    . Software: HBO & Company
    Proprietary/Maximumps

CHOOSING A NEW VENDOR
THE SELECTION PROCESS

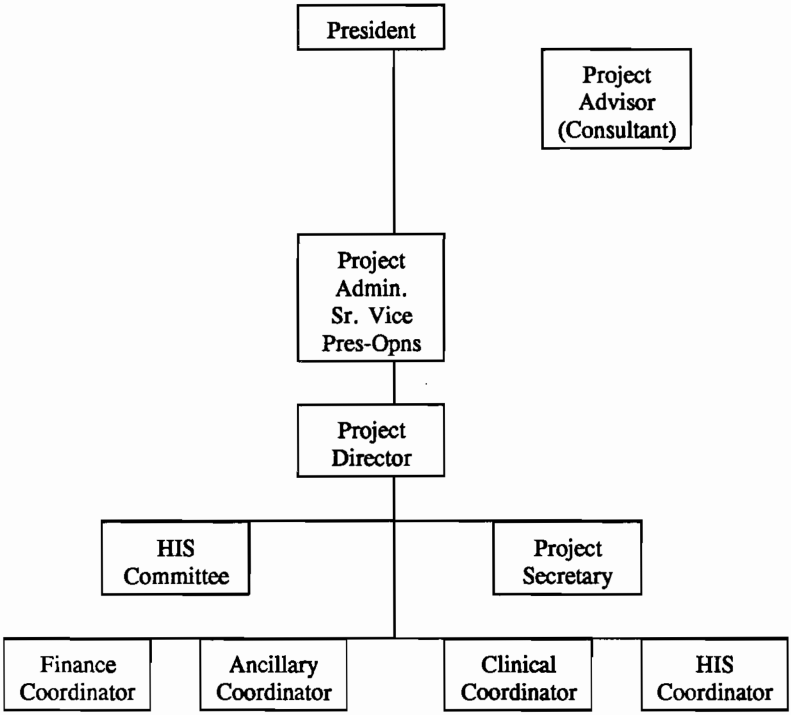
It was determined by Senior Management early in this process to enter into a contract with a consulting firm that had previously done a major study at the hospital. Since we had been satisfied with their work and they were already familiar with our institution, it was decided to retain this firm for this project. The scope of the consultant's work would be to . . .

1) Develop the Hospital-Wide Benefits Needs Analysis

2) Prepare the Request For Proposal (RFP)

3) Participate in the vendor selection

As the project began, a task force of approximately twelve (12) was appointed. This task force consisted of a wide cross section of management personnel from the various hospital departments (Finance, Senior Management, Nursing, Radiology, Laboratory, Pharmacy and Information Systems). Appointed by Senior Management to chair this group as the Project Director was the Director of Pharmacy.


President
Project Advisor (Consultant)
Project Admin. Sr. Vice Pres-Opns
Project Director
HIS Committee
Project Secretary
Finance Coordinator
Ancillary Coordinator
Clinical Coordinator
HIS Coordinator

The consultant guided the HIS committee through the review process of the current systems on the market. The committee determined that the vendors to be evaluated would be no more than six and that site visits would only be done with two vendors.

After the vendor market was reviewed, five vendors were selected. These vendors were requested to complete a very lengthy Request For Proposal (RFP) and on-site demonstration were arranged for all of them over a 2-3 month period. The vendors evaluated during this part of the process were:

DataCare
Gerber Alley
HBO & Co.
Meditech
Unisys

Following the demos, the dinners, the pro-ball games, etc. the final two vendors were selected and were identified as Gerber Alley and HBO & Co (the incumbent).

Next came the corporate jets, trips to California, more dinners and some very long days. We were presented with interfaces, integration, multi-vendor solutions, open architecture, proprietary systems, maximumps, better solutions, the only solutions, "vapor ware," etc. The RFP's were in, the site visits were completed and now came the final vote. And the rest is history for us at BMMC --the vendor of choice became Gerber Alley -- a young health care vendor with no known Radiology product and some missing pieces. But the fit was right; we liked what we heard; we wanted freedom to control our output; we wanted standard operating system. This was only the beginning and the hard work was still to come. It was now September of 1989 and we had a planned go-live of July 1, 1990.

CONTRACT NEGOTIATIONS/CONSULTANT INVOLVEMENT

Having selected a vendor of choice, the next step was to negotiate a contract. We were determined that we would not accept any vendor's "standard contract." As an institution we knew there were certain applications we had to have up and running to replace existing systems but we wanted to "reserve" the right to purchase the additional applications at a fixed cost and not the market price at the time the decision to install would be made.

BMMC also wanted to include in the contract certain vendor requirements and of course the vendor had their list of responsibilities. Contractually we wanted: ----

- Response Time
- Fees/Payment Schedule
- Volumes
- Custom Programming
- Escrow Agreement
- Implementation Plan
- Hardware Definitions
- Software Definitions

The consultant's involvement through this process was extremely important and in the final analysis, we were able to negotiate a "good" contract.

COMMITMENT/TRAINING

A massive task now loomed before us. That responsibility would now focus on getting the commitment of the entire hospital. This meant all levels had to be committed because there would be overtime involved, time away from employee's assigned tasks and dealing with change throughout the institution.

In addition, we were faced with two (2) contracts that would expire in October 1990 for our existing systems and we did not desire to extend these contracts if at all possible. So a "go live" date was set as July 1, 1990 and meetings were scheduled in the hospital to explain our plan of action to all employees.

Job descriptions for three new positions were written and the openings posted. Only a few people applied but we were able to selected some outstanding people. These people came with the desire to make things happen and were challenged by change. The Clinical Coordinator position was filled by the former Assistant Director of Nursing for Oncology services; the Financial Coordinator position was filled by the former Administrative Office Manager; the Ancillary Coordinator was filled by a former Cytotechnologist from the Laboratory. All these people became part of the staff of Information Systems but for the period of implementation reported directly to the Project Director.

After the Project Team was selected, the Training Rooms were identified; hardware was ordered; Definition Workshops were set; Implementation Schedules were prepared and published and the consultant involvement would be only in an advisory capacity. This would be OUR SYSTEM. We would be responsible as a Project Team from the beginning and would continue to support this system and future applications as they would be installed.

IMPLEMENTATION SCHEDULE

Phase/Task Description	Dates
Management & planning	September 1989
Definition/analysis	October 1989 - December 1989
Build	January 1990 - February 1990
Training	February 1990 - June 1990
Verification	October 1989 February 1990 - June 1990
Conversion/live	April 1990 - June 1990
Production	July 1990 - August 1990
Hardware	March 1990 - June 1990
Anticipated Go Live July 1, 1990	

CONVERSIONS

Nothing causes more difficulty in making system change than conversions of data from one system to another. If there were any way to avoid data conversions, new system implementation would be significantly easier. The reality of the situation is, however, that no company can avoid converting certain information. Our file conversions consisted of:

Accounts Receivable	45,000 open accounts
Patient History/Demographics	400,000+ records
Payroll/Human Resources	2,100 employees
General Ledger	2,800 accounts

Even though our original intent was to also convert our Radiology Film History information, another solution resulted in placing this information on four (4) stand-a-lone PCs with excellent results at a very low cost. Since our existing systems were "interfaced" and not "integrated" the result of trying to convert both the Patient History and the Radiology Film History into one conversion could have been disastrous.

In the health care environment, systems of the future will become even more complex to master conversions due to the magnitude of installed applications (Laboratory, Radiology, Pharmacy) and the installation of integrated systems sharing common data bases.

CHANGE TO THE INSTITUTION

Much publicity and fanfare preceded our July 1 "go-live." This was a very important part of the process and helped to keep the interest level high throughout the institution. Not only does the installation and implementation of a system of this magnitude involve numerous departments, it also involves volunteers, clergy, on-call staff, employees on leave of absence, etc.

During the training process we had our "live" data base as well as a "train" data base which allowed the users to train and practice not only in the designated training rooms but also at their work station since several weeks prior to the anticipated live date all hardware was placed on the units and department work stations. In the "live" data base the I.S. Department was doing the testing and conversions of data from our current systems in preparation for the actual go-live date.

Dealing with change at this level is a monumental task. We not only had to deal with how we process information but at the same time we were responsible for patients' lives. We had to insure that systems were working properly and that departments were receiving orders and that results were being transmitted back to the units. The software vendor supplied numerous people for staffing our hotline around the clock. The I.S. operations staff was buried in paper trying to get accustomed to the new reports that came out of the nightly batch and trying to meet users requests to make changes to the reports at the last minute. Even though we had run batch in the train data base for many weeks prior to the actual live date, the shock of actually going live and making sure that the output was correct and distribution lists were updated and personnel became familiar with the system, there were many difficult moments and long hours by all the staff.

Our support from the Senior Management was outstanding (and very forgiving) and the team effort on the part of all the staff of BMMC was most commendable.

Signs were posted at key patient areas informing them of our new system and asking for their patience in the registration process. Were there long lines? YES. Did we lose any employees over this change? YES. Did we have any upset patients? YES. But lots of positive reinforcement on the part of the Project Team members and quick problem resolution resulted in a very successful implementation without any major setbacks. There was little finger pointing and we all knew we were in this together and it was summer time in Michigan and we all longed for some time off to enjoy the great Michigan weather. There was significant "let down" experienced on the part of the Project Team within a few weeks. We had worked hard for these past months and we were driven with the anticipation of going live on the new system. After a few days the software support people from the vendor returned home and within a few weeks the Project Team didn't feel as needed as before. We had to be challenged again -- make rounds, ask questions, seek quick problem resolution and develop a new mode of operation. We continue to meet routinely as a project team for the purpose of continuing the communication and support of our HIS.

CHALLENGE FOR THE FUTURE

The challenge to BMMC came in July 1991 when we went live with the Laboratory Management system from the same vendor. We anticipate enhancing the Laboratory system by adding the following modules:

Outreach
Direct print (fax) of Lab results to outreach facilities
Blood Bank
Anatomical Pathology

What does the future hold for other applications? We are very much interested in adding the following modules:

Patient Care Plans
Resource Scheduling System
Clinical Data View of Patient Results/Drugs
Executive Information System
Inquizitive (User Report Writer)
Automated Third Party Claims Submission

Our Commitment and challenge at BMMC is to have an integrated system. We feel that we are on the way to achieving this. We are not there yet but for management reporting we are only as far away as the creativity of the manager. We still struggle with the daily responsibilities of operating a Information Systems Department and handling all the trouble calls. Also like all of you, we have limited resources to accomplish our goals. We remain excited about the future and when we actually hear over the intercom in a physician's office a physician say, "please check the Blodgett system to obtain the results of Mrs. Smith's radiology tests of this morning an print them for me please" -- then we really get excited because we know we are also providing a wonderful service to our physicians in their own offices and improving the quality of patient care. At the same time we eliminate telephone calls to our various hospital departments resulting in improved hospital productivity.

REALIZED SYSTEM BENEFITS:

Regulatory Compliance
Access to Data
Elimination of Manual Tasks
Improved Public Image
Improved Data Analysis
Single Access Point for Users
Report Writer Capabilities
Centralization of Patient Care Information
Improved Nightly Batch Time
Improved System Security
System Integration

INSTALLED HARDWARE:

(1) Hewlett Packard Model 980/200
(10) 571Mb Eagle Drives
(12) 1.34Gb FL Disk Drives
(3) 798OXC Tape Drives
(1) 300 LPM System Printer
(1) 600 LPM System Printer
(1) Laser Jet 2000 Printer
(245) HP 700/92 Terminals
(120) TI and HP Application Printers
(70) PC Devices Attached Via 3Com PC Network (Reflection)
Dawning Laboratory Instrument Interfaces
Intermec Bar Code Readers used in Radiology & Laboratory

THIRD PARTY SOFTWARE:

Adager
LaserSoft from Unison
QEdit from Robelle Consulting
Powerhouse from Cognos including:
- QUIZ
- QUICK
- QTP
- GRAPHICS
- POWERPLAN

Reveal from O'Pin Systems
DataExpress

PAPER NO.: 6006

TITLE: Business Intelligence Application in a Client/Server Environment

AUTHOR: Michael Cassettari
Information Resources, Inc.
200 Fifth Avenue
Waltham, MA 02154
617-890-1100

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO.: 6008

TITLE: Helping Paper Addicts Go Cold Turkey or Profiting From Electronic Reporting in the Information Age

AUTHOR: Kevin Mitchell
O'PIN Systems
7900 International Drive #635
Bloomington, MN 55425
612-854-3360

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO.: 6010

TITLE: Business Intelligence Systems - The Ultimate in Integration

AUTHOR: Mark Symonds
IISI
63 Nahatan Street
Norwood, MA 02062
617-769-7511

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

Paper #6011
What Exactly is an EIS?

Matt Henson
CMSI
2901 Squalicum Parkway
Bellingham, WA 98225-1898
(206) 734-5400 ext. 2414

Introduction

Executive Information Systems (EIS) seems to be one of the buzzwords for the 90's. A great deal of time and energy is spent writing about EIS, but almost no one bothers to stop and define the terms. Is an EIS the same as an ESS? as DSS? What are the similarities and differences? Why should we care?

This paper is meant to serve as an introduction to the concept of EIS, and will define and describe the common approaches, user requirements, technical requirements, benefits and problems associated with EIS in today's business environment.

What is an Executive Information System?

Simply stated, an EIS is a computer-based system designed for executives to gain insight into and to keep track of critical success factors via easy-to-use interfaces. An EIS should be able to incorporate the unique decision-making style of an executive and supply the information necessary to monitor the executive's critical success factors in a timely and accurate fashion.

Approaches to EIS range from mainframe-based systems, to networked PCs and minicomputers, to stand-alone PCs. The most common approach is to use PCs connected to host mainframe or minicomputer. Information is gathered and manipulated on the host, and the PC is used as a user-friendly workstation. Information may be downloaded to the PC ahead of time, or may be sent to the PC as the information is requested.

This kind of computer usage has the potential to redefine how corporations are managed. Daily monitoring of business functions is now possible, allowing for more timely correction. Many companies are willing to take the risk of introducing new technology to remain competitive. A well-built EIS application can serve as an early warning system that helps executives pinpoint problems and track their competitors' moves in the marketplace.

Background

The phrase 'executive information system' was first used in a 1982 Harvard Business Review article written by John Rockart and Michael Treacy. Rockart, the director for the Center for Information Systems Research at M.I.T., had been performing research into the information requirements of senior corporate executives. What he found was that most of these executives do not have immediate, timely access to the types of information they require to make decisions.

Additional research into this area yielded a few companies whose executives had systems designed specifically for their information needs. He found that these systems could be generalized to fit a simple model. These systems shared three characteristics:

1. *a central purpose.* The top executives who personally use computers do so as part of the planning and control processes in their organizations. The reason for EIS systems is to support more effective use of information by executives.

2. *a common core of data.* Although no two EIS systems are identical, each contains information on important business variables (also referred to as 'critical success factors'). What sets this information apart is the breadth of its sources and the depth of its detail. In many cases, this data is complemented by information on major competitors, key customers, and important industry segments.

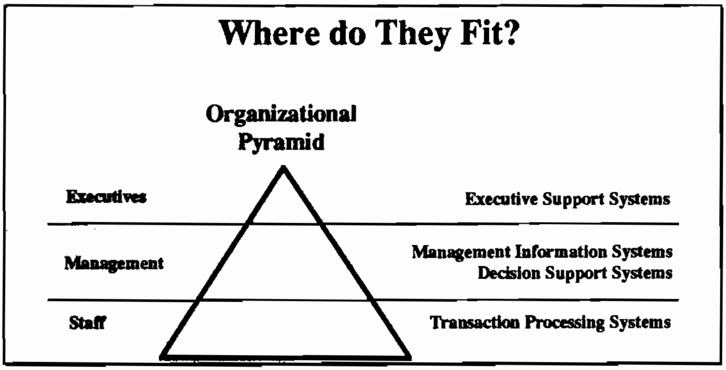
3. *a support organization.* All EIS systems observed in Rockart's study depended on a high level of one-on-one support for the executive users. This support is essential if the system is to demonstrate its full potential. Executive users have little time or patience to learn new systems. They will require training, and will need help in conceptualizing and improving their systems and analyses. This function is best performed by people who have excellent people skills,

an understanding of systems, and an understanding of the underlying business concepts.

"There is a huge advantage to the CEO to get his hands dirty in the data, because the answers to many significant questions are found in the detail," according to Rockart and Treacy. "The EIS system provides them with an improved ability to ask the right questions and know the wrong answers."

How does an EIS differ from Other Type of Systems?

Organization are often characterized as pyramids, with the executive level at the top, management levels in the middle, and clerical and production staff at the bottom. One way to classify information systems is to associate them with the level of the organization pyramid which they support.



Transaction Processing Systems

Transaction processing systems are aimed at the lower levels of the pyramid. Originally, they were designed to mechanize the work of clerical personnel. These systems typically support specific functional areas (such as payroll or accounts receivables,) and attempt to make these functional areas more efficient.

Management Information Systems

These systems (MIS) historically arose from budgeting and cost accounting systems. They are targeted for use by middle and lower management. Historically, these systems produced large volumes of printed output and included various levels of summarization (i.e. COBOL break reports). The current trends in this arena are towards on-line storage of summarized data and reports, and on-line inquiry and ad hoc reporting.

Executive Support Systems

Executive support systems (ESS) are systems designed to support the specific needs of executives. Rockart and Delong's definition of ESS is broad and encompassing:

> "The routine use of a computer-based system, most often through direct access to a terminal or personal computer, for any business function. The users are either the CEO or a member of the senior management team reporting directly to him or her. Executive support systems can be implemented at the corporate or divisional level. Such a system does not require hands-on access to the technology and has been designed expressly for the senior manager. Further, unlike DSS, which offered few text capabilities, ESS enhance senior managers' ability to communicate clearly, via electronic mail and bulletin boards, for example."

In "Is your ESS meeting the need?", Gary Gulden and Douglas Ewers break down ESS systems into four categories based on the concepts of simplification of information access, motivation of management performance, acceleration of communication, and expansion of insight and understanding. Four types of ESS arise from this classification schema, Office Support Systems, Management Control Systems, Decision Support Systems, and Executive Information Systems.

ESS vs. EIS

ESS: A Variety of Executive Support

EIS: Simplificationof Information Access

DSS: Expansion of Insight and Understanding

MCS: Motivation of Management Performance

OSS: Acceleration of Communication

Office Support Systems. These systems facilitate accelerated communication through the use of electronic messaging, voice mail, calendaring, word processing and imaging. Implemented correctly, they remove timing and location constraints and integrate electronic messaging and office automation to increase productivity.

Management Control Systems. These systems facilitate organizational change by measuring and reporting key features of management performance indicators required to meet organizational goals. Measurement of key performance indicators becomes simpler through automation and make the information available for the appropriate levels of management to review.

Decision Support Systems. Decision support expands the insight and understanding of the business by providing a reporting and trend analysis system which does not require programming. This results in accelerated availability and communication of this information. These systems are usually based on historical information, but are often focused on the forecasting and analysis of the impacts of changes. In addition, DSS often help identify opportunities for change by highlighting trends or other useful information.

Decision support systems are an outgrowth of management information systems. These systems provide the functions of management information systems as well as additional features such as trend analysis, modeling and "what if" scenario generation, and graphical and textual reporting capabilities without additional programming efforts. They systems provide tactical support for management. Decision support implies the use of computers to: (1) Assist managers in their decision processes in semistructured tasks; (2) Support, rather than replace, managerial judgment; and (3) improve the effectiveness of decision making, rather than increasing its efficiency. A typical DSS has a command-line interface which is used to enter commands via the keyboard. The commands are then acted on by the software.

Executive Information Systems. EIS attempt to provide executives with timely access to historical information through simplification of access, thus enabling them to expand their insight and understanding. Usually such systems use graphical user interfaces (GUIs) and devices such as a mouse, track ball or touch-screen displays. These systems attempt to make existing information more readily accessible to top-level executives. A full-scale EIS includes information generated outside of the organization regarding the industry, the competition, and technical innovations. The internal information presented tends to begin at high level summaries, and enables the executive to "drill down" or "peel the onion", progressing through various levels of detail. Then, by using exception reporting, it can assist executives in identifying problems in critical areas. Thus, the primary purpose of the EIS is to turn data into information, and make it easily accessible.

User Requirements of an EIS

Critical Success Factors. First and foremost, an EIS is designed to provide executives with information that helps them monitor the factors which they view as critical. If the information provided by the EIS is not in sync with the executives' needs, the system will fail. This information should be presented from a "top-down" point of view. Summary information is presented first, and the executive can choose which details require further review.

Ease of Use. Because an executive has even less time available to spend at the computer than other users, the user interface of an EIS must be extremely "user-friendly". To accomplish this, dependence on the keyboard is minimized through the use of pointing devices such as a mouse, light pen or touch screen. GUIs, featuring icons and color, are used extensively to communicate ideas, especially in menu selection and reporting. In addition, on-line context-sensitive help screens should be provided so that the user can get relevant help quickly.

The following example was offered by an EIS builder for a chemical company: "We want our plant managers to have a system that shows real-time information on the status of each part of their operation. The best interface is a diagram of the process. Parts of the diagram should change color depending on current data. A problem area, for instance, should flash red. Then the system should allow the plant managers to point at and select those items they want to review. The EIS should instantly display the information describing the performance of that component."

Some interesting experiments with user interfaces have taken place in the efforts to sell EIS to top executives. In one case, a corporation founded by ex-Air Force pilots used a set of dials, reminiscent of a cockpit layout, to display the key indicators of the business. Another scheme which has been successfully used is to display a matrix of green, amber, and red squares, highlighting the areas that need management attention relative to some performance standards.

Graphics and Reporting Flexibility. Executives (like most other users) are impatient. They want systems that allow them to choose the graphics format they like best. EIS that don't offer that flexibility can come under heavy fire.

Speed. An EIS must be fast. Experienced EIS directors say that new reports must appear in less than two (2) seconds and new graphs in less than six (6) seconds. Systems that take as little as thirty (30) seconds per display are ridiculed for their molasses-like quality.

Multiple Methods for Locating Information. Because of the amount of information available through an EIS, it must support multiple methods of locating information. Executives will not tolerate being forced to traverse multiple layers of icons or menus to reach the report they want. Instead, they need key word searches, pictorial indexes, and other methods which allow them to see the information they want to find quickly and efficiently.

EIS most often provide information categorized in many ways, including strategic business unit, functional area, key performance indicator, product and location. Each of these is often using much of the same basic data, simply divided and viewed differently.

Exception Reporting. When reviewing graphs or tables of internally-generated information, the EIS should report potential problems through the use of exception reporting. This is done by internalizing pre-defined performance standards and conditions into the EIS programs. Colors are frequently used to highlight the severity of problems.

For example, when reviewing annual employee turnover, the organization may consider 20% turnover as a severe problem, and 10% turnover as less severe but noteworthy. Managers whose departments have experienced high turnover might appear in red (problem), and moderate turnover might appear in yellow (noteworthy). Managers with acceptable turnover levels might appear in green.

Drill-Down Capability. When the executive user would like to see more detailed information on a specific segment of a graph, or a line in a table, they simply point at the data in question and "drill-down" to the next level of detail available. This can be repeated through several layers of information until the nature of the problem becomes evident.

In the turnover example mentioned above, the summary level turnover information would show individual managers, each of which may or may not be responsible for multiple departments. Drill-down capabilities would allow the executive to identify which of the manager's departments are experiencing turnover problems, and which are not.

Hard Copy Output and Data Export. After the executive finds the data which s/he needs, often they require the ability to print or graph it. Because of this, hard copy output (preferably in color) is required. In addition, the software should be able to export information in a format which can be used in popular programs such spreadsheets, word processors or graphics programs. This allows people to use the information communicated through the EIS in memos, reports and ways the developers hadn't anticipated.

Access to External Information. One of the features which makes EIS software unique is the ability to access externally generated information. According to a University of Georgia study, the primary external sources include new services, stock markets, and trade/industry data. Ideally, the EIS will have the ability to access several of these services and will electronically scan for pertinent information that the executive needs to be aware of. Usually, this scanning is accomplished through the use of keywords, such as competitor or client names, industry buzzwords, etc. In this way, the executive can review only those articles s/he is most likely to be interested in. This feature alone can save a great deal of time.

Other services, such as publicly available financial data, and private database may also be of use to the executive. Automated interfaces and background request processing make these sources of information available with a minimum of delay.

Office Automation Support. Although often considered a separate module for ESS, features such as electronic mail, calendaring, group scheduling, imaging and word processing are features that enhance the ability to communicate internally. When implemented and used throughout an organization, these features can eliminate the problem of "phone tag" to discuss important information or schedule meetings involving several individuals. The ability to attach a graph or article to an interoffice memo is also a desirable feature.

Completing the earlier turnover example, the executive could attach the table which reveals the turnover problem in a memo to the manager requesting an explanation as to why the turnover is occurring. In addition, s/he could send a copy of the memo to the human resources department for further review and follow-up.

Technical Requirements for an EIS development system

Most EIS products do not provide a turnkey system for management. Instead, they provide a shell for building applications. The shell manages the information to be displayed, refreshing it as necessary, and provides the interfaces and retrieval facilities needed. In the July 16, 1990 edition of Computerworld, EIS development systems were evaluated and reviewed.

Products selected for review had to be "full-capability EIS", which means they had to be designed for use by top-level management and offer the following functions:

Data Access. Because the source data for the EIS may exist in numerous systems, the EIS development platform should be able to integrate data from a wide variety of hardware and software platforms. It must efficiently extract data from internal databases, as well as provide access to external databases. It should also include tools which facilitate the creation of multilevel ad hoc queries on the data passed to the EIS.

Application Development Tools. Historically, graphical applications have been very difficult to implement. However, when a senior executives wants to see a new graph or report, they expect timely results. Therefore, the development platform must support rapid development and easy maintenance of applications. The platform must provide for prototype development, preferably through the use of an application generator or CASE tool. These generated screens and reports must also be easy to customize. In addition, effective support for multiple user interfaces, including a keyboard, mouse or touch screens, is a required feature.

Excellent Technical Support. Quality of service and technical support are critical features. As a rule, these systems are complex and expensive. If problems occur, the vendor must have an excellent support organization available at the technical and managerial levels to help resolve problems quickly.

Communication. In order to access external databases, new services, and interface with office automation software, the development platform must support a wide variety of communications protocols. It should also support remote access, since executives and management staff are often not in the same physical locations.

> "EIS is definitely [meant to be] a networked product," say David Friend, cofounder and CEO of Pilot Executive Software. "To buy an EIS and assume you can plus it in and put it on one person's PC doesn't do much good because an EIS has to be an information resource. To that extent, it should be shared by a lot of people to coordinate tracking and control of business practices. They have to be tied together and have one centralized source of information. That's the whole point."

Security. The development system must provide mechanisms for tight security of data, screens and reports. Because the EIS provides sensitive information, more than two-thirds of the firms questioned by a University of

Georgia study said they employ some combination of user identification, password and device security.

Benefits of using an EIS

> "EIS is one of the most important product categories for the 90's because it provides people access to the information they want without forcing them to go through the data processing activities," said Alan Paller, well known EIS author and consultant.

Unlike traditional information systems, EIS provide no clear, easily defined cost savings. In fact, few systems would survive a traditional cost-benefit study in straight labor saving terms. Often, there is no simple way to quantify the value of a better or speedier business decision. If a decision needs to be made within a given time frame, it is made. A successful EIS will streamline the decision making process by providing more easily accessible information, thus improving the quality of the resulting decision.

There are four advantages unique to EIS that more and more executives find appealing.

1. EIS systems offer significant assistance to analytically-oriented executives in their search for a deeper understanding of their companies and industries.

2. EIS systems can be structured to accommodate the information needs of the individual manager.

3. EIS systems can start small, providing support to one executive in a single area. It can then evolve as others become interested, adding the data sets and access methods appropriate to each new user. As the system grows, synergy is attained because the information requirements of different executives in the same organization tend to overlap, and growth occurs only as additional users "buy in".

4. EIS facilitate and compensate for reductions in staff and middle management. In many companies, middle management acts as a filter for information, and this causes delays. By implementing a system that facilitates the transfer of information without direct management intervention, the span of control for a given manager can be increased.

Problems of Implementing and Using an EIS

Politics. With a doubt, the biggest problem encountered in an EIS project is the political one. In every major corporation, there are certain organizations or individuals who control the destiny of the company by controlling the flow and availability of information. Often problems never reach the executives who should be handling them. Instead, they are handled by lower-level power barons, who decide what information should be fed up the corporate ladder and what information should be quietly ignored.

An EIS causes a major shakeup in the flow of information. Middle managers lose their control over data. Information now flows from the analyst or database directly into the systems for distribution to all top executives. Since middle managers no longer have autonomous control over the content and distribution of the data, it becomes much harder to hide problems. When implementing an EIS, anticipate organizational resistance because an EIS is inherently political. Changing information flows and access is very threatening to middle managers.

Hardware, Software and Implementation Costs. The system purchase price is just the down payment. Regardless of the initial price, the total expense to implement an EIS will far surpass the initial investment in hardware and software. These combined costs will probably represent only 20 to 50% of the total systems costs. Personnel development costs account for the majority of the remaining expenses.

The biggest hidden costs stem from the need to restructure data and modify systems so that previously unrelated information can be brought together and presented on the executive's desk. Organizations using highly integrated systems will find that EIS implementation is much simpler than those with multiple, disparate sources of the same information.

Hidden costs can also flow from customizing systems for executives. The time that executives spend helping to design and learning to use the system must be considered. Often, an EIS system can create a whole family of supporting information systems, simply because top management starts to look at the data.

Empirical Bias. Business communities tend to share the empirical bias of science. Accompanying this empirical bias in business is the ever increasing availability of data delivered by EIS and DSS. Together, they threaten the value of management decisions by providing empirical data in a form which is easily used by executives. The danger associated with these systems is that they are, as all computer systems are, model-based. How is the executive going to know when the model is wrong?

When data is gathered for analysis and models built, the users of the models must possess the experience to identify critical aspects of the process that are not incorporated into the model. Experience is a guide to roughly what range of outcomes to expect before an analysis is run. Abstractions in models don't always act in the same way as the real world components they purport to represent. An unexpected analytical results leads an experienced manager to suspect the model first and their internal sense of the process last. The opposite tends to happen to an inexperienced manager. Prior knowledge of the processes producing the data being analyzed, and how well the indicators being tracked actually capture the fundamentals of those processes, helps an experienced manager to resituate the findings of an analysis in the real world.

Therefore, an EIS that isolates management from the underlying reality has the potential to lead them astray. Executives cannot allow the information generated by these packages to become the only reality that they see. They cannot afford to lose touch with the processes that generate the data.

Summary

ESS and EIS systems hold a great deal of promise as vehicles which can deliver information efficiently to top executives. Even for companies who cannot afford full blown implementations, the concepts embodied in this class of software should provide direction for informed information systems professionals and industry software vendors.

> As one CFO put it, "Why spend hundreds of thousands of dollars to put technology in front of clerks who earn five dollars an hour, but not in front of executives who earn more than $100,00 annually?"

For full-scale EIS implementations, there are many products currently available. Some provide tool sets, others provide "canned" versions of EIS. More and more vendors are offering EIS solutions with turnkey software. In the healthcare field alone, there are several turnkey offerings for EIS, but none of these constitute full-scale ESS, including integrated EIS, DSS, Office Automation, and Management Control Systems. Expected costs for full-scale EIS implementation start at approximately $50,000 and increase according to the complexity of the desired results and environment.

For small to mid-range EIS implementations, various personal computer based systems are now available. With the use of the Macintosh or Windows, most of the interface requirements can easily be met. As usual, software is lagging behind hardware. However, during the last couple of years, EIS development platforms are increasingly functional and inexpensive. (Two example platforms which retail for less than $1,000 are Channel Computing's Forest and Trees Lan Version and Cognos' PowerPlay.) As PC software gains functionality and ease of use, and hardware continues to gain speed, software packages which deliver the kind of flexibility and ease of use will be increasingly available. These graphical database reporting and analysis tools should provide excellent solutions when teamed with other ESS components available for these operating systems.

About the Author:

Matt Henson is employed as a Systems/Project Manager for CMSI, a facilities management and consulting firm based in Portland, OR. He is currently working with St. Joseph Hospital, a 250-bed acute care facility in Bellingham, WA. Matt has an MBA and an undergraduate degree in Computer Science from Western Washington University.

Bibliography

Alexander, Michael. "Exec info systems catch on" Computerworld, vol 23, Feb 27, 1989, pp 31-33.

Armstrong, David. "The people factor in EIS success" Datamation, vol 36, Apr 1, 1990, pp 73-77.

Booker, Ellis. "A simple yet elegant route to EIS success" Computerworld, vol 24, May 7, 1990, pg 35.

Brightbill, Tim. "Executive info system value rests on clean data delivery" Healthweek, Jan 28, 1991, pp 23-24.

Burke, Steve. "IBM readies EIS products" PC Week, vol 6, May 22, 1989, pg 11.

Carlson, Eileen. "What color is your EIS?" Computerworld, vol 24, Jul 16, 1990, pp 79-82.

Caroll, Paul B. "Software helps managers find the data they need" The Wall Street Journal, Oct 1, 1990, pg B1, col 2.

Churbuck, David. "Next time, think big" Forbes, vol 143, Jun 12, 1989, pp 155-156.

Coffee, Peter. "Pilot's Lightship is rich, sweet but unfulfilling" PC Week, vol 7, Oct 15, 1990, pp 33-34.

David, Stephen G. "Can stand-alone EISs stand up?" Datamation, vol 35, Jul 1, 1989, pp 41-43.

Eskow, Dennis. "ITT Avionics revamps EIS to measure work quality" PC Week, vol 7, Aug 6, 1990, pp 121-123.

Eskow, Dennis. "Michigan State Senate elects an EIS" PC Week, vol 7, Oct 8, 1990, pg 159.

Eskow, Dennis. "Phasing out of middle managers may increase dependence on EIS" PC Week, vol 6, Aug 21, 1989, pp 71-72.

Ferranti, Marc. "NewsEdge service gains link to Pilot Command Center" PC Week, vol 7, Jun 4, 1990, pg 40.

Ferranti, Marc. "Pilot aims Windows-based EIS at non-programmers" PC Week, vol 7, Sep 10, 1990, pp 39-40.

Fireworker, Robert B. and Zirkel, William. "Designing an EIS in a multidivisional environment" Journal of Systems Management, vol 41, Feb 1990, pp 25-31.

Flaaten, Per O., McCubbrey, Donald J., O'Riordan, P. Declan, and Burgess, Keith. Foundations of Business Systems. Chicago:Dryden Press, 1989.

Frenkel, Karen A. "The war for the executive desktop" Personal Computing, vol 14, Apr 27, 1990, pp 56-62.

Gulden, Gary K. and Ewers, Douglas E. "Is your ESS meeting the need?" Computerworld, vol 23, Jul 10, 1989, pp 85-89.

Hedley, Rees-Evans. "Top management transformed as EIS arrives" Accountancy, vol 103, Apr 1989, pp 143-144.

House, William C. and Napier, Herman S. "Computers in the executive suite: acceptance is slow" The Office, vol 110, Oct 1989, pp 28-29.

Keen, Peter G. W. and Scott Morton, Michael S. Decision Support Systems: An Organizational Perspective. Menlo Park:Addison-Wesley, 1978.

Kelly, John N. "Executive Information Systems" Patricia Seybold's Office Computing Report, Vol 11, No 12, Dec 1988.

Lucas Jr., Henry C. Information Systems Concepts for Management. New York:McGraw-Hill, 1990.

Ludlum, David. "System purchase price is just the down payment" Computerworld, vol 23, Dec 4, 1989, pg 94.

Main, Jeremy. "At Last, Software CEO's Can Use" Fortune, Mar 13, 1989, pg 77.

Martin, James. "DSS applications should shed new light on a problem" PC Week, vol 6, May 1, 1989, pg 50.

Martin, James. "EIS helps managers gain insight into factors for success" PC Week, vol 6, Apr 24, 1989, pg 54.

McCartney, Laton. "How ESS keeps Hertz managers out in front" Business Month, vol 134, Jul 1989, pp 46-47.

McNamara, Brian, Danziger, George, & Barton, Edwin. "An appraisal of executive information and decision support systems." Journal of Systems Management, vol 41, May 1990, pp 14-18.

Moran, Robert. "MSA announces EIS started kit at a 'smart' price" Computerworld, vol 23, Jun 19, 1989, pg 16.

Nelson, Robin. "Culture clash at the top (MIS and top executives often conflict on development of executive information systems)" Personal Computing, vol 14, Apr 27, 1990, pp 76-77.

Pallatto, John. "$495 EIS package helps users monitor firms financial health" PC Week, vol 7, Jun 11, 1990, pp 51-52.

Paller, Alan and Laska, Richard. "How PCs can do the job" Personal Computing, vol 14, Apr 27, 1990, pp 67-70.

Paller, Alan and Laska, Richard. The EIS Book Homewood, IL:Dow Jones-Irwin, 1988.

Picarilla, Lisa. "SPC plans low-cost, OS/2-based EIS program" PC Week, vol 7, Jul 23, 1990, pg 4.

Rockart, John. "The Changing Role of the Information Systems Executive" Sloan Management Review, vol 24, Fall 1982, pp 3-13.

Rockart, John and Delong, David. "ESS shakes up the system" Computerworld, vol 22, Feb 29, 1988. pg 58.

Rockart, John and Delong, David. Executive Support Systems. Homewood, IL:Dow Jones-Irwin, 1988.

Rockart, John and Short, James E. "IT in the 1990s: managing organizational interdependence" Sloan Management Review, vol 30, Winter, 1989, pp 7-17.

Rockart, John and Treacy, M.E. "The CEO goes on-line" Harvard Business Review, vol 60, Jan-Feb 1982, pp 82-88.

Ryan, Alan J. "Analog introduces twist to executive information system" Computerworld, vol 24, Feb 5, 1990, pg 30.

Ryan, Alan J. "Cost of EIS a big deal for most firms" Computerworld, vol 23, Jul 24, 1989, pg 46.

Ryan, Alan J. "Making room at the top for EIS" Computerworld, vol 24, Feb 26, 1990, pg 14.

Schier, Robert L. "Bank enlists EIS in battle to boost control" PC Week, vol 7, Feb 5, 1990, pp 129-130.

Sexton, Tara. "Host graphics spur hospital's EIS" PC Week, vol 6, Jul 3, 1989, pp 37-38.

Shea, Kelly. "Getting it right the first time" Computerworld, vol 23, Jul 10, 1989, pp 89-90.

Sullivan, Kristina B. "Touch screens ease EIS access to corporate data" PC Week, vol 6, Sep 25, 1989, pg 98.

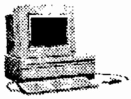
Sullivan-Trainor, Michael L. "Command Center slips past Commander (barely)" Computerworld, vol 24, Jul 16, 1990, pp 88-91.

Watson, Hugh and Glover, Harry. "Common and avoidable causes of EIS failure" Computerworld, vol 23, Dec 4, 1989, pp 90-91.

Watson, Hugh. "Avoiding hidden EIS pitfalls" Computerworld, vol 24, Jun 24, 1990, pp 87-89.

Handouts for Paper #6011, What Exactly is an EIS?

EXECUTIVE INFORMATION SYSTEMS



1

What is an EIS?

- Critical Success Factors
- Top Down Interrogation
- Easy to Use

2

Implementation Approaches

Mini/Mainframe

Client/Server

Stand alone PCs

3

Historical Review

Rockart & DeLong, HBR 1982

Central Purpose

Common Core of Data

Support Organization

4

Where do They Fit?

5

EIS vs ESS

ESS: A Variety of Executive Support

EIS: Simplification of Information Access

DSS: Expansion of Insight and Understanding

MCS: Motivation of Management Performance

OSS: Acceleration of Communication

6

Handouts for Paper #6011, What Exactly is an EIS?

User Requirements (1 of 2)

- Identify Critical Success Factors
- Ease of Use
- Graphics & Reporting Flexibility
- Speed
- Drill-Down Data Interrogation

Presentation by James M. Henson - Paper 6011 7

User Requirements (2 of 2)

- Exception Reporting
- Multiple Methods of Locating Information
- Hard Copy Output/Data Export
- Access to External Information
- Office Automation Support

Presentation by James M. Henson - Paper 6011 8

Technical Requirements

Data Access

Application Development Tools

Flexible Communications Capabilities

Security

Presentation by James M. Henson - Paper 6011 9

Benefits of an EIS

- Can Assist Analytically Oriented Executives
- Can Accommodate a Manager's Unique Needs
- Can Start Small and Expand
- Can Help Reduce Middle Management Control over Data and Provide Upper-Level Execs with a more Accurate Picture of the Organization

Presentation by James M. Henson - Paper 6011 10

Problems of an EIS

Politics

Implementation Costs

Empirical Bias

Modeling Problems

Presentation by James M. Henson - Paper 6011 11

Conclusions

"EIS is one of the most important product categories for the 90's because it provides people access to the information they want without forcing them to go through the data processing activities."

Alan Paller-EIS Author & Consultant

Presentation by James M. Henson - Paper 6011 13

PAPER NO.: 6012

TITLE: EDI Basics: Mapping, Translation and Communication

AUTHOR: Larry W. King
Computer Management Solutions
7208 South Tucson Way #175
Englewood, CO 80112
303-790-0123

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO.: 6013

TITLE: Developing Quick Response EDI Software

AUTHOR: Rod E. Keiser
Computer Management Solutions
7208 South Tucson Way #175
Englewood, CO 80112
303-790-0123

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

Paper #6015

What To Do For An Encore

Roy N. Dailey
Assured Systems Inc.
984 North Broadway
Yonkers, NY 10701
(914)963-4401

PREAMBLE

Systems are changing and the task of managing systems is changing as well. The maturation of the computer industry has left successful professionals looking for the next level of production systems. What do we do for an Encore?

INTRODUCTION

Today the majority of us are working with stable systems that handle routine data processing. They work. We are quite proud of what we have created, designed, installed or managed. But computers can do more. Most companies have yet to apply technology effectively to aid executive decision making. We have served the clerical but not the cerebral side of our organizations.

As one cynic observed, "More and more data is being gathered with less and less purpose." There has been a tremendous explosion in the availability of information. Simultaneously, the widespread availability and use of Personal Computers, along with modestly priced yet sophisticated software has improved computer literacy and awareness. Knowledgeable users have generated scores of ideas and requirements.

Tom Peters, of "In Search of Excellence" fame, talks today of FICKLE systems, systems that can be quickly and easily changed to provide continuously relevant information in a rapidly changing commercial world.

We have to prepare for flexibility in the future without disturbing the firm foundation of today's systems.

CHARACTERISTICS OF CHANGE

New subsystems must share the following characteristics:

- no major changes to existing systems
- no system downtime
- low cost
- minimal technical skills to implement/install
- user friendly interface
- no visible performance degradation

To meet these needs we should:

- make extensive use of package software
- build small but sophisticated bridges between package products
- avoid building major custom systems
- separate data processing and information processing

Recognizing the distinction between data processing and information processing is the key to developing our Encore. Maintaining this separation allows us to start building the supple, flexible, and easily changed information systems needed in the executive suite, without disturbing the basic business systems.

CHARACTERISTICS OF DATA PROCESSING & INFORMATION PROCESSING

DATA PROCESSING PROJECTS

- low risk of failure
- easily measured or visible payoff
- limited payoff value
- raw data entered (ORDERS, INVENTORY, PAYROLL)

INFORMATION PROCESSING PROJECTS

- high risk of failure
- hard to measure or estimate pay off
- instinctive high payoff value
- usually requested by a senior manager/executive
- driven by output of data processing

When I say risk of failure, I don't mean that programs fail but that the results did not help the user as anticipated. Surgery was successful but the patient died! Data processing professionals do not willingly start projects that might fail. It is not our environment. However, we must adapt to survive.

The following examples illustrate historical data processing and information processing systems. All but one are taken from my own experiences.

EXAMPLES OF DATA PROCESSING PROJECTS

FAX

One client, a New York distributor, had for a long time been "faxing" Purchase Orders to major suppliers. You can imagine the scene. Someone searched anxiously at the printer for relevant PO's then beat a well worn path to the fax machine and tediously transmitted each document. A very obvious improvement would be to automate the fax transmissions. This, of course, is what we did. We used a dedicated PC set up with a FAX board and appropriate software linked serially to the HP3000. The PC interrogates the HP on a regularly scheduled basis looking for FAX material. When found, the FAX file is automatically downloaded and transmitted.

These two programs are the "Bridge". The original PO system was modified at the print stage to create a disk file instead of a spool file for FAX vendors, a minor change.

This low cost solution requires no operator intervention and uses mainly off the shelf components.

EDI

In the commercial world more and more large companies are insisting that their suppliers transmit documents via EDI. The virtue of EDI is that standard document formats can be electronically transmitted, routed, and received among trading partners. Any dominant company which can establish EDI with all its trading partners realizes tremendous benefits. This has been well documented, and successful case studies abound.

The system we had to implement was for a manufacturer with over 10000 active customers who was required by one to transmit invoices via EDI. Here EDI was more difficult to justify, except as an accommodation to a large, influential customer. The solution chosen was a PC based EDI system linked serially to the HP3000. The PC interrogates the HP on a regular basis looking for EDI invoices. When found the EDI file is downloaded and subsequently transmitted.

Again the "Bridge" was created, and minor changes were made to the standard invoicing system.

The operation is unattended and, therefore, cost effective on a day to day basis.

CREDIT/COLLECTION INFORMATION

This example highlights a technique of adding significant new functionality to an existing system without jeopardizing its integrity. Support from the original software vendor is, therefore, maintained.

The requirement was to provide online tools to help the collection process. Such a system is not hard to conceive. The credit department must have access to the latest accounting data and the ability to restrict credit when necessary. In addition the credit department needs an online log of its own credit/collecting activities, a tickler file and the ability to generate letters.

The solution is to implement the changes while insulating the source data. In this case we built a complete subsystem with its own database and only read access to the production database.

There is some duplication of fields between the two databases, but this has been a small price to pay to maintain the original systems integrity.

ARCHIVAL RETRIEVAL

An impressive name hides a simple technique. Many production databases are cluttered with out of date material being saved for future potential analysis. Data that is too important to purge, yet not needed online. The cost of retention in the current production database is response time degradation, increased backup time and larger dataset sizes.

What we can do is to keep a mirror image of the database structure. Periodically, we extract non current data from the production database and load it on the mirror image.

The setup cost is low because we use existing utility software.

Furthermore, the same programs used for retrieval on the production database are used on the mirror database.

In this way we maintain a lean fast production database and a large, slow archival database. Often this leads to dramatic performance improvements.

EXAMPLES OF INFORMATION PROCESSING PROJECTS

Information Processing is the pot of gold! All our equipment, systems and staff have been cost justified by the success of Data Processing. However, here are some examples where successful Information Processing has produced dramatic benefits to the organizations involved.

LOAD MANAGEMENT

This one we have all read about. American Airlines Load Management was an information system added to the original SABRE Reservations System. SABRE itself is just a sophisticated order entry and inventory system with the inventory being a highly perishable commodity - passenger seats. From the raw historical inventory data, an Information System evolved tracking flight loads. This information enabled one man to start forecasting probable loads, then to adjust loadings by juggling prices. This fine tuning of pricing and load has proven a major triumph.

I remember reading that the President of the Airline attributes a significant portion of ticket revenue to Load Management.

QUALITY-CONTROL

This example was also based on an order entry/inventory system. This time the inventory items were computer main frame components and peripherals. The manufacturer (famous for building VECTRA clones!) recorded customer sales using a standard Data Processing System. From this raw data we built an Information System designed to check hardware compatibility. The Information System checked simple things, like all power options being of the same voltage, or that each disk subsystem was equipped with at least one controller and disk channel. This quality control Information System then became a marketing tool for highlighting customers who would need upgrades. These projections were fed back to the factories for very reliable production forecasts. The financial value was clearly significant.

PRODUCT SALES

This client set up annual sales comparison by class of product for each salesman. The original intent was just to review the contribution of each product class to corporate profitability. However, the data was very skewed, particularly for one salesman who had no sales for some very popular lines. Discreet inquires revealed that he represented another supplier as an active sideline! The outcome of this Information System was identifying the inventory laggards and highlighting the benefits of different salesmen's techniques. The company credits this system with several points of sales gain.

NATIONAL SALES

Another client examined annual sales by customer. He used the information to eliminate expensive sales calls to unprofitable accounts. He transferred such customers to a telephone order entry system. Not only did this save money but it improved customer service.

INVENTORY USAGE

Since these are high risk projects, I must tell you of another client where we set up an information system to forecast inventory usage and establish reorder points. The forecasting model is very sophisticated, and switches between several time series techniques, as well as, multivariate regression techniques. Unfortunately, the results have been less than helpful, and we have virtually abandoned the project.

SUMMARY

In the early stages of an Information Processing System we must bear in mind:

- each system is tailor made to an individual user
- each system must conform to the user's personal style - not everyone wants to use a spreadsheet or manipulate a keyboard
- very few users realize how much data is available
- very few users know how the computer can be used to extend their creative endeavors
- never to underestimate the user's intellectual ability
- always be conscious of cost so that abandoning a project is easy
- the information system must always be independent of basic data processing systems to provide flexibility

We have to be very sensitive to the reactions of our users, perhaps more so than with a traditional data processing systems. Here are some of the more significant user comments about Information Systems:

- ease of use not as important as ease of remembering
- independence, access when convenient
- ability to interact not react
- benefits of timely online data
- online information/results cannot be edited/slanted or filtered

CONCLUSION

An Encore is the time to be versatile. Our opportunity is Information Systems. Here are the seven critical ingredients:

<u>Sponsor:</u> Every successful project needs one individual to take personal responsibility.

<u>Urgency:</u> The need must be urgent and the pay off dramatic.

<u>Commitment:</u> Management must be committed to face the challenge.

<u>Clarity:</u> We need a clearly defined set of objectives, and established criteria for success.

<u>Education:</u> We must teach about tools and data available, and also learn from the user. Communication is critical.

<u>Setup Plan:</u> We must establish milestones and budgets.

<u>Strength:</u> We need the strength and fortitude to handle hurdles which may be financial, technical, cultural or political.

Sponsor

Urgency

Commitment

Clarity

Education

Setup

Strength

"What To Do For An Encore"
6015-10

PAPER NO.: 6016

TITLE: Teaching a Dinasour to Play Hop-Scotch

AUTHOR: Dick Biggs
FMC Corp. - Conveyor Equip. Div.
P.O. Box 1370
Tupelo, MS 38802
601-869-7558

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING.

PAPER NO.: 6501

TITLE: HP Systems: The Best Mainframe Alternative

AUTHOR: Robert Winter
Hewlett-Packard Co.
c/o Ella Washington
19091 Pruneridge Avenue, M/S 46LG
Cupertino, CA 95014
408-447-1053

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO.: 6502

TITLE: HP 3000 Open Systems and Client/Server Computing

AUTHOR: Christine Fronczak
Hewlett-Packard Co.
c/o Ella Washington
19091 Pruneridge Avenue, M/S 46LG
Cupertino, CA 95014
408-447-1053

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

Paper No: 6801

Implementing EDI: Three Steps to Heaven?

Trevor Richards
M.B.Foster Associates
9417 Great Hills Trail
Suite 2038
Austin, TX 78759
Tel: 512 345 5376

In the words of the song 'there are three steps to heaven ... just follow steps one, two and three'. To derive the maximum benefit from Electronic Data Interchange (EDI), there are three clearly defined steps which need to be taken. Companies which embark on full implementation of EDI will find that their businesses are radically different when these three steps have been followed. This article outlines the three implementation steps necessary to achieve maximum return for your EDI investment, the advantages and disadvantages of various implementation options, and the benefits that can be achieved after each step. A real life example of business reorganization, resulting from the successful implementation of each of the three steps, will then be used to demonstrate how EDI can modify both the internal and external business practices of each trading partner.

What is EDI ?

EDI has been defined in many different ways over the past two decades, and the definition is continually being refined to accommodate the novel applications in which this technology is being employed. EDI is currently defined as 'the automated computer application to computer application exchange of structured business documents between an enterprise and its trading partners'. When the term "trading partner" was originally adopted it referred to either a customer or a vendor of the company in question. Today, the term has been expanded to include any entity with whom a company exchanges business documents. In the very near future this will include federal and state government, health care providers, insurance companies, the IRS, etc. "Structured business documents" generally comprise of a "transaction set" constructed according to a particular "standard". A standard is a method of structuring business documents in an electronic form, which has been agreed at company, industry, national or international level. A transaction set is the electronic form of a group of similar business documents, such as a group of purchase orders, which have been structured according to an EDI standard.

To fully understand the benefits that EDI can offer, it is necessary to first describe the problems which it was intended to solve. The traditional method for communicating business documents is the mail system. This medium is slow, it typically takes 4-6 days for business documents to be exchanged. When the business documents are received, they are typically rekeyed into the receiving party's computer application. It is also highly likely that the documents were printed from a computer application at the sending party's site. If documents are sent via the mail system, it can take 6-10 days to transfer them from the sending party's computer application to that of the receiving party. The telephone was the first medium to be used to try to reduce the time taken to exchange business transactions. However, the telephone is an imprecise medium. The process of the caller reading details to the recipient who in turn types those details into a data entry screen, is a scenario which can all too often lead to errors in the final business document. Overnight delivery and facsimile are recent developments which can both cut down considerably the cycle time for the exchange of business documents. However, overnight delivery is costly, and both mediums suffer from the fact that the documents have to be rekeyed into the receiver's computer application.

These cycle times were acceptable when businesses were run with large inventories, but when techniques to minimize the inventory held began to be employed it was critical to find ways to reduce this limiting factor. Just In Time (JIT), in particular, is a technology in which the cycle time needs to be as short as possible. The implementation of JIT results in reduced batch sizes for manufacturing or distribution companies, and orders are placed more frequently. The increased volume of business documents which need to be exchanged between trading partners, together with the much shorter cycle time required for each individual transaction, caused large companies, particularly those in the automotive and retail industries, to look for a more efficient and timely way to exchange business documents. The solution that they adopted was to exchange business documents in a standardized electronic form.

In the 1960s, the large automotive and retail companies defined proprietary fixed length file formats for the exchange of business documents. Trading partners of these companies were thus forced to support several different proprietary formats. In the 1970s, the transportation industries began the use of an industry standard for the electronic exchange of data, and defined a data structure for their EDI standard which was the foundation for all of today's standards. The Transportation Data Co-ordinating Committee (TDCC) defined transactions comprising of variable length fields, or data elements, in variable length records, or segments. This was done to minimize the number of characters which needed to be transmitted, but introduced a need for software to 'translate' this format into fixed length records, and vice versa. The grocery industry and the warehousing industry also introduced their own EDI standards, UCS and WINS, during this period. The 1980s have been typified by the use of national EDI standards such as that maintained by the ANSI Accredited Standards Committee X12 for the USA. Most company standards have been replaced by, or are in the process of being replaced by, transactions within the ANSI ASC X12 standard. Industries are now typically recommending their members to implement specific X12 transaction sets according to their industry's implementation guidelines, rather than developing new standards. Even the transportation, grocery and warehousing industries are in the process of redefining their transactions under the X12 standard. The 1990s are likely to be typified by a move from national standards to the single international standard EDIFACT, which should facilitate international trade.

Many transaction sets have now been defined under the X12 standard and cover business areas as diverse as quality data reporting through tax returns. The most comprehensively covered business area is the purchasing function, with the price/sales catalog (832), the request for quote (840), the purchase order (850), the purchase order acknowledgement (855), the ship notice (856), the invoice (810) and the remittance advice 820).

Trading partners rarely communicate EDI transactions directly, preferring to route them through Value Added Networks (VANS). VANS offer a store and forward mail box service, which allows trading partners to send and receive EDI transaction sets when their schedule permits and without concern over their partners' schedules. Thus, the entire EDI process involves trading partner A extracting transactions, such as purchase orders, from an application, translating them into EDI transaction sets and sending them to a VAN. Trading partner B then picks up the transactions, translates them into a format that his application can read, and loads them into the receiving application.

What are the benefits of implementing EDI ?
A survey of the benefits of EDI was conducted by ANSI Accredited Standards Committee X12. Companies who had implemented EDI were asked to rank various benefits according to their own experience. Surprisingly, cost saving was placed only seventh in the list. The top five benefits were improving relations with trading partners, maintaining competitiveness, improving quality of service, reducing cycle time and improving competitive advantage. The fact that these less tangible benefits were placed much higher than cost savings could be due to a number of factors. It is likely though that many of the participants in the survey have only just begun to implement EDI and are in the period where the costs of implementing EDI have not been exceeded by the cost savings it can bring. The cost savings that EDI can offer are proportional to the extent to which EDI has been integrated with the various functions of a business. The three steps to full EDI implementation will now be described.

Step 1 : Communications
In this first step, business documents are received from and/or transmitted to trading partner(s) without integration to any existing computer applications. Many companies begin their EDI implementation in this way by installing a stand-alone personal computer with a PC-based EDI software package. To receive transactions from trading partners, the package must be capable of communicating with a VAN, downloading the EDI transactions, translating the EDI transactions into a fixed length file, and printing the transactions, such as

sales orders, in a human readable format. The printed copy of transactions received would then be used to update the appropriate computer application, such as a sales order processing system. To send transactions, such as sales invoices, to trading partners, the package must have a data entry capability to allow transactions printed on hard copy from the computer application to be rekeyed into the PC software. The transactions entered are then translated into EDI transactions and sent to the VAN.

This is by far the most common entry point for new users of EDI and there are several advantages that this route offers. Firstly, it offers a relatively low cost entry into EDI with PC-based software ranging from $500 to $3000. If there is business pressure from large customers to implement EDI, then this route offers a solution that can be implemented quickly and which offers a fast EDI learning curve. If there is no internal computer application to receive or transmit the transactions to be exchanged, this is probably the only possible solution. This step offers improvements in most of the metrics used to gauge EDI benefits but is hindered by the need to rekey documents into receiving computer applications or from transmitting applications. There is consequently still a time delay in rekeying incoming documents, such as sales orders, together with a similar error rate for the rekeying operation as that encountered for paper documents received via traditional methods. For transactions to be transmitted, there is even an additional data entry operation introduced by the need to rekey documents printed from an internal computer application. This method of implementation is appropriate if there is a relatively low volume of transactions sent and/or received, and there is external pressure to implement some form of EDI quickly. It should be borne in mind, however, that the only part of this initial EDI investment which is likely to be carried forward to steps 2 and 3 is the EDI education which your staff have been given.

Step 2 : Integration
In the second step, business documents are received from and/or transmitted to trading partner(s) with complete integration to existing computer applications. Many people would argue that this is "real EDI", where

business documents are transported through a standardized format from one trading partner's computer application to another's computer application without human intervention. In addition to the communications and translation functions mentioned above, there are two additional functions that are necessary to allow EDI transactions to be tightly integrated with application systems. Firstly, EDI software needs to provide a "mapping function" to reformat EDI transactions into a fixed length file format consisting of transaction header, detail and trailer records which will be intelligible to an application system, and vice versa. Application systems in turn need to be able to provide "application interfaces" which either load a fixed length file produced by the EDI software package through a batch interface, or allow an extract file to be created for those transactions which need to be sent via EDI rather than by paper document. The application interfaces are generally created and maintained on the host computer on which the application system resides. However, the functions of communications, translation and mapping can reside on either a PC front-end or on the host computer.

A PC front-end is a much lower cost option than a host-based solution. Prices for EDI software with a mapping capability on a PC are generally in the $2000 to $6000 range. With the appropriate communications between the PC and host in place, a PC front-end can provide an integrated solution for EDI, which leaves the host relatively unaffected by the processing requirements of EDI. However, the PC front-end may have a much lower processing capability than the host system to which it is attached, and can end up being a "bottle neck". The PC front-end also places another link in the chain, from the trading partner's application to the receiving application, which can fail.

The main advantage of implementing EDI software on the same host processor, as the application system which is the target or source of the transactions to be exchanged, is that it minimizes the number of linkages between the trading partner and the application system. There is generally more in-house technical expertise in both data communications and operating system support on the host computer, and the host can process much larger volumes of data. The main disadvantage of host-based EDI software is the cost, which is typically in the range from $10,000 to $40,000 depending on the features offered. There is also some concern over the additional processing load that EDI software places on the host computer.

Integration provides the solutions to all of the problems which were identified earlier with the traditional methods of communicating business documents, so what more can EDI do for you?

Step 3: Optimization
The accuracy and timeliness of EDI transactions allows companies to be much more creative in the way that they, and their trading partners, conduct their businesses. The optimization phase of EDI implementation involves the radical change of existing business practices to maximize the benefits associated with the use of EDI technology. This statement sounds all well and good, but what does it mean in practice ?

Integration of EDI transactions with application systems merely replicates our existing paper-based environment in a much more efficient manner. There is still the same amount of data passing back and forth between trading partners as there was before. Companies who had integrated EDI into their overall business systems decided to look more closely at the transactions which they were transmitting and receiving. A study of the data held in successive documents in the purchasing cycle shows that each document contains 80-90% of the data which was included in the preceding document. So, not only is there much of the information that is passed between trading partners that duplicates data previously transmitted or received, but that each trading partner is also validating this information before loading the transactions into the receiving application system. Consequently, the automotive industry has embarked on a process of re-evaluation of the transactions which pass between customer and vendor. The business document which they first identified as being "redundant" was the invoice. The price has previously been agreed when the purchase order was placed, and this is generally the only additional information carried by the invoice over that contained in the ship notice. To remove the invoice from the purchasing cycle, both customer and vendor need to modify their business practices.

This is illustrated by looking at the traditional settlement process which was used in General Motors. The ship notice and invoice were sent to GM via EDI, and were then forwarded to separate departments. The receiving department get a copy of the ship notice, and accounts payable received a copy of the invoice. When the goods were received, the receiving department would send a receiver document to accounts payable. Accounts payable would then match the receiver document with the invoice, and authorize payment of the invoice. The remittance details would then be transmitted via EDI to the vendor. If the price has been previously agreed, and has been accurately established in both the customer's and vendor's computers via EDI, the traditional function of "checking the invoice price" that accounts payable performed is redundant. If the price is not an issue upon receipt, the receipt of goods in accordance with the issued release schedule can drive payment of the vendor. General Motors now employs the evaluated receipts settlement process. The vendor sends an advanced ship notice via EDI, which contains details of the bar coded shipment and package labels which have been attached to the shipment just leaving the loading bay. The ship notice is passed electronically to receiving for reconciliation with the incoming shipment. When the shipment arrives, the bar coded shipment label is read and the advanced ship notice details sent via EDI are recalled. As the bar code label on each package is read, the shipment details are reconciled. Once the shipment has been successfully reconciled, an evaluated receipts settlement transaction is actioned, which will generate payment for the goods to the vendor once the agreed terms of payment have been reached.

Summary

The benefits to be obtained from EDI are large, but the changes demanded of a company and its supporting business application systems are equally daunting. The message for MIS professionals is clear : to reap these rewards application systems, whether acquired from a third party or developed in-house, will need to at least provide interfaces for EDI transactions, but in the long term will need to be flexible enough to support the changes in common business practice brought about by the novel application of EDI to business problems.

#6804 Considerations, Experiences and Process for Developing an Expert System Prototype

L. Davis and A. Bykat
The University of Tennessee
Chattanooga, TN 37403
(615) 755 - 4165

1.0 Overview

The purpose of this presentation is to discuss the process of building an expert system prototype. The expert system is the vehicle that carries the presentation not the object of the presentation. Our actual experiences and considerations that transcend the specifics of either the application or system are the primary discussion points.

The authors built an expert system prototype to aid in the academic advisement of students at UTC. The primary goal was to construct a software program that would advise students as to what courses students should take in their next semester of study. The expert system developed took into consideration both those courses typically taken by a student in a given year and term and as well a student's specific academic successes such as passing or failing a required calculus course. A secondary goal was that by building this first system as a prototype, much could be learned as to how to construct a system more complete in features and services. This initial prototype could be scrapped and redone more efficiently and easily rather than reworking the original. The final goal was to evaluate the concept of a shell against that of a programming language such as PROLOG for building such a system.

A knowledge engineer, skilled in developing expert systems, worked with a domain expert, skilled in academic advising. They developed, using a PC based shell, a prototype expert system which is both knowledge based and consultation system. This prototype was quickly built to test the system and concept.

2.0 Background

Information Management has many technological tools that hold promise for the next decade. These include to name a few: CASE, Object Oriented Programming, and Expert Systems. The latter, expert systems, is one of the most useful applications that emanates from the more general topic of Artificial Intelligence. Expert systems may be defined as software that utilize knowledge or rules of operation that are supplied by a human who is expert in some field. These expert systems model the decision making process, advising, or other thought processes that the human expert demonstrates as the individual goes about his or her typical duties. Several of the more well known include Delta by GE in the 80s to identify and correct maintenance problems in diesel locomotives, Mycin by Stanford University in the early 70s for use in diagnosing and prescribing treatment for certain bacterial infections, and XSEL by DEC for configuring VAX computer systems. Most involve a series of complex if then rules, coupled with a large store of facts that could bear upon a solution, and the intimate and detailed knowledge about the thought processes of humans as they go about finding a solution without the aid of a computer. The goal of these expert systems is twofold: to analyze data and problems and to recommend to the less expert user probable solutions or action plans.

Until recently expert systems were found largely on mainframe computers since such systems required large amounts of memory, high mip computing power and often large data bases. Further the software and hardware resources required were expensive and could not easily be justified for small or lower cost computing hardware environments. However expert systems engines or shells now can be purchased for the price of a word processor or spreadsheet and can operate on systems as small as 386 hardware, or perhaps even on a 286. Rather than programming an expert system in a programming language per se such as Prolog or Lisp, shells offer an attractive alternative. Expert system shells utilize natural language interfaces to input the expert's application data, rules, and knowledge to form the knowledge base. The shell contains as well the software (inference engine) that runs the consultation for the user. Additionally there are typically editors, print and trace utilities, and both help and debug facilities within these shells.

The specific expert system shell that we utilized at the University of Tennessee at Chattanooga was Texas Instruments Personal Consultant (TM) Easy Version 2.0, 1986, 1987. This particular package was purchased in 1987 as a site license for our campus. PC Easy is touted by its manufacturer as... "PC Easy is a highly functional tool for developing small or medium-sized expert systems. PC Easy is also well suited for rapidly creating prototypes and for teaching or learning about expert systems."

3.0 The Problem

Teaching, research, and public services are the foundation stones on which higher educational institutions are built. The first of these, teaching, has a concomitant responsibility for advising students on their program of studies. In today's world of education, few students finish their program of studies leading to an academic program with no disruptions, detours, hiatuses, or false starts. When educational programs are scaled back due to financial pressures, students with educational backgrounds that are incomplete or insufficient for their program of study find that faculty advisement is a necessary condition for academic success. Thus the problem is to build a system that will take a student where s(he) is academically and advise them as to what are the courses that should be taken next that most expedites their degree aspirations. In the language of critical path, find an optimal (minimal time) program of study leading to the sought after degree. That is the purpose of the expert system prototype built by the authors for use at the University of Tennessee at Chattanooga.

4.0 Prototype

For purposes of a prototype, the authors determined to investigate how to implement an expert system for the B.S. in Education: Secondary Mathematics. One author is in fact the academic advisor for students in that major. In a given semester there are 40 to 50 advisees. Actually the goal was made somewhat smaller. We decided to concentrate on the mathematics course sequences, the education courses and several other topics that students might plausibly be taking as a sophomore in such a program. Hence we prototyped the mathematics, education, and several other courses for students in the BS in Education: Secondary Mathematics. These are the students preparing to teach mathematics grades 7 to 12. In actuality many students enter the program unable to begin the Calculus sequence Math 150. Others transfer in from other majors or from other institutions and are thus out of step with the nominal program given in the University Catalog. All students do not complete their mathematics courses with Cs or better. This is compounded by the necessary prerequisite structure of mathematics and hence its largely sequential nature. Students can often take only one course in a given semester and rarely take three mathematics courses in a given term. Hence a student at any level can in practice be taking required mathematics courses a year or more above or more often below their academic level or year. A nominal four year program follows:

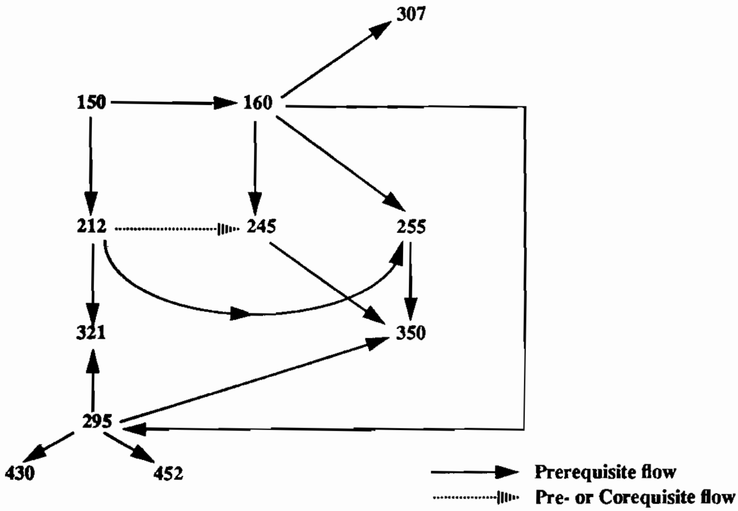
Typical Course of Study in Secondary Mathematics (B.S.):

	Fall		*Spring*	
Freshman				
	English 121	3	English 122	3
	Mathematics F150	4	Mathematics 160	4
	Physical Education	1	Gen. Educ. Category C	3
	University Studies 101	3	Theatre & Speech 107 or 109	3
	Psychology C101	3	Science Elective D	4
	Computer Science 102	3		
Sophomore				
	Mathematics 212	3	Mathematics 245	3
	Mathematics 295	3	Gen. Educ. Category B	3
	Philosophy B201	3	ED Curr. & Inst 204	3
	ED Curr. & Inst. 201	3	Mathematics 307	3
	Physics D230	4	Physics 231	4
Junior				
	ED Curr. & Inst 321	3	Mathematics 321	3
	Mathematics 255	3	Mathematics 452	3
	Mathematics 430	3	Physical Education Activity	1
	Humanities/Fine Arts Elec.	3	Gen. Educ. Category B	3
	Computer Science 121	3	ED Spec. Ed. 333	3
	Gen. Educ. Category G	3	Mathematics Elective	3
Senior				
	ED Curr. & Inst. 433	4	ED. Curr. & Inst. 438	12
	ED. Curr. & Inst. 431	3		
	Mathematics 350	3		
	Mathematics Elective	3		

5.0 Process

The mathematics advisor was to be the domain expert. The other faculty member, a professor in the Computer Science Department and Senior Fellow in the Center of Excellence for Computer Applications was to act as the knowledge engineer and technical advisor on expert systems. The PC Easy software was installed on a PS/2 Model 55 and the domain expert started a study program with a set of tutorials on PC Easy by T.I. Based on consultations between the two authors, an area of mathematics education was determined to be both key to the advising and representative of the problem, hence worthy of prototyping. This was the mathematics courses themselves, the education courses, and a few other courses for good measure. Goals were established for the program to be considered successful. The domain expert plotted first the sequencing of the mathematics courses both required and elective. This sequencing included both the term and year that the student had academically achieved and the typical courses for a student at that stage. Additionally the requisite and co-requisite structure of these courses was plotted. The same process was followed in Education. See Figure1 for the mathematics structure.

FIGURE 1. Core Math Requirement



Next the PC Easy System was started in a development mode. The goals to be achieved were entered in the Goals Section. The parameters to be used during a consultation session were entered. Finally the systems knowledge base (KB) was created by entering the rules. Initially the rules were entered for the sophomore year, then freshman, and concluded with junior and senior. After each year's rules were entered, the logic of these rules was examined by running a test consultation and confirming the expected results. Debugging tools such as Trace and Why were used to explain unexpected results.

Essentially the logic showed that certain classifying decisions were of major importance, such as the importance of Math 212 and Math 295 and later both Math 245 and Math 255. The logic was then redrawn from that analysis and a new KB-Rules set entered. Even that had problems, since the sequencing is not solely requisite driven. The logic was reentered to modify the latest KB-Rules to influence course selection also by what is considered to be the most appropriate course for a given student at a given time. As an example Math 350 can be taken early in the junior year if looked at from prerequisite standards but from academic maturity it is best taken in the senior year. This third rewrite of the KB-Rules seemed to be satisfactory. We then concluded by adding parameters and rules for English, University Studies, and Physical Education course requirements.

6.0 Caveat

Development of an expert system demands a programming mind set much different to that produced by standard programming classes. Programming (developing) an expert system is an exercise in declarative programming (and in a strongly constrained mode at that). By comparison, programming in languages such as Pascal, Fortran or Cobol demands a procedural approach. Conceptually, the difference between the declarative and procedural approaches can be signified briefly as the difference between specifying how to achieve a result and stating what the result should be. Procedural approach specifies the mechanism for solving a problem, whereas in declarative programming a mechanism exists and relations to be used by the mechanism are stated.

Whereas programming using the declarative approach is easier, the learning curve for this approach is much steeper than that for the procedural approach.

7.0 Expert System structure considerations

As indicated above developing an expert system is an exercise in declarative programming. Conceptually, an expert system consist of a knowledge base, a data (facts) base, and a solution mechanism [Weiss, 1984; Waterman 1986]. The solution mechanism (referred to sometimes as an inference engine) searches for and invokes appropriate rules. The data base includes facts - evidence for the rules - provided by the user for a specific case being solved, as well as facts derived by applying the rules to that specific case. The knowledge base consists of a set of rules, which are used to verify and to derive facts, and which a domain expert provides as valid for the problem domain.

The solution mechanism is essentially a symbolic search and match mechanism which scans the knowledge base for rules which are applicable to a specific case (in the problem domain) submitted by the user for solution. Since the rules are presented as if-then (situation-action) rules, application of a rule corresponds to verification that the situation prevails, and subsequently to action being taken. The action is typically an assertion of new facts, or invocation of additional rules.

There are two principle approaches to evaluation of applicable rules: backward chaining and forward chaining. Many of the expert system shells provide the backward chaining mechanism. Some, e.g. CLIPS provide the forward chaining only. Yet others, as is the case with PC Easy, provide both mechanisms for expert system construction.

Backward chaining is a goal directed approach. In this approach the user poses goals and the system attempts to verify/satisfy these goals by identifying subgoals that need to be verified first. Recognizing that subgoals are goals in their own right, this process is then applied recursively, until all subgoals that remain to be verified are facts. If these facts are known to the system to be true, the subgoals and thence the original goals, are then concluded as verified/satisfied. In the backward chaining approach, an if-then rule is interpreted as goals-subgoals rule, where the *then* part represents goals and the *if* part represents subgoals.

Forward chaining is conceptually the inverse of the backward chaining process. It is a data (facts) directed approach, in which the system attempts to verify facts and if the latter are found to be known to the system as true, the appropriate rule is then invoked. In this approach the *then* part is considered to represent the facts describing a situation, and the *if* part presents rules to be invoked and additional facts to be established.

It is a premise of the expert system methodology, that human experts employ rules in performance of their tasks. Frequently these rules are heuristic. Such rules are undoubtedly synthesized by the expert using his conceptual understanding of the physical system under consideration. This conceptual understanding is the expert's model of the system. When incorporated as a set of rules into an expert system, it constitutes a *static implicit model* of the physical system. Static expert systems use rules formulated by human experts, abstracted from the underlying model, and presented as the (software) knowledge base. This abstraction process does not preserve the conceptual model as a separate entity; the model is present *within* the rules in a highly compiled and therefore highly inflexible state. In such - first generation - expert systems [e.g. Bykat, 1989; Bykat 1990], change of the physical system may imply significant changes to the rule base and fact base. Nonetheless, the solution mechanism remains unchanged, and consequently any changes to the expert system would pail in comparison to changes that a corresponding procedural system (if the problem was so programmed) would require.

Most of the current work on knowledge-based systems is based on the assumption of a static expert knowledge base representing implicitly the underlying physical system. Such implicit representation is highly compiled and can be highly efficient (though inflexible to changes). The point to note is that the static expert system captures the knowledge of the system at a certain point in time, but does not support evolution of the system. The rules in such a system match current data against predetermined static situations.

Currently emerging second generation expert systems attempt to overcome this drawback through provision of an explicit model [Bykat, 1991]. These systems may cope with physical system evolution by permitting dynamic modification of the knowledge base.

8.0 Experiences

The major effort in developing an expert system is specification of the relationships governing the domain of the problem. These relationships are typically stated as facts and rules. Facts, in PC Easy, correspond to values of constants and variables, whereas rules are stated in the if-then (logical implication) form.

The following is an example of a rule used in our prototype, which utilizes a number of facts. In particular the rule states that if a student is currently a sophomore, junior, or senior (not a freshman) and has not taken Education 201 then the education course most important (goal) for the student to take is Education 201.

FIGURE 2. Math Advisor Rule (Rule 020).

If ! e201 and year= sophomore or year= junior or year= senior

Then educ= "EDCI 201 unless you are still a Freshman"

While specification of the relationships is a major development effort, selection of a suitable domain is of critical importance. Selection is typically based on experience of lessons learned, and mistakes made, in specification of relationships. Of course, a number of guiding principles can be posed for particular shells. These will arise from capabilities (or lack of thereof) of the particular expert system shell. For example, capabilities such as ability to deduce multiple solutions, ability to preserve derived solutions and use of those in subsequent manipulations, ability to deal with multiple valued variables, etc. would play a role in selection of a suitable problem for expert system approach, and a suitable shell for its expert system development.

Further, general principles, independent of particular shell selection, can also be posed. For example, problems which can be solved by formulation and consideration of a hierarchical classification are eminently suitable for expert system approach.

8.1 Experience with PC Easy

An expert system shell such as PC Easy offers a number of advantages:

- the ability to describe exactly the outcome needed for specific existing conditions, or courses, e.g. "Take EDCI 201 if you haven't already and you are beyond your freshman year."
- the use of a structure into which goals, parameters, and rules are built.
- an Editor within the shell to modify, add, or erase components such as knowledge based rules.

- debugging tools such as Trace, Why, and How to explain logic flow, for what reasons information is requested, and the logic as to how a decision is reached.
- consultation software to do the actual student advising.
- multiple forms by which information is input to the expert system and is internal to the system to read data that is single or multiple valued variables, choosing one or more choices from a menu, or yes or no responses.
- both backward and forward logic chaining.
- ability to upload and download data to a database.
- management of files, directories, and utilities within the shell.

PC Easy has difficulty in a number of areas:

- multiple values - whereas multiple values are available, extraction of their individual components is not straight forward.
- backtracking - failing a rule does not induce back tracking, thus evaluation of all possible solutions is difficult.
- setting a general database, (e.g. course prerequisite database), building and preserving a data base of courses taken and grades achieved (in order to verify prerequisite sequences as well as to reduce advisement effort) requires knowledge of another software package (e.g. DB3).
- preserving values of recommendations for students for tracking specific student and reuse of that information in subsequent advising consultations with the same student.
- data structures (beyond number and range) are not supported. This makes expression of relationships between various attributes (name, subject, courses taken, passing grade) of courses taken difficult.
- promotes the construction of long complex rules as the logical consequences of the decision tree being modeled.
- sequencing of rules is set by the order in which they are entered (modifications can alter this) and can not be easily altered or resequenced.
- no easy way to control goal seeking using incrementation or summation.
- access to Scheme (a simple dialect of Lisp) is not readily available from the shell. Such access would be needed to overcome many of the above difficulties (e.g. modify the expert system rule hierarchy).

9.0 Prolog, an alternative

Prolog - Programming in Logic - is a general declarative language which offers capabilities that make it an alternative to programming an expert system using an expert system shell. Prolog has a very simple formalism, coupled with a very powerful search mechanism based on a backtracking and back search approach. Prolog is based on predicate logic, and has risen to become a major AI programming language on a par (and competing with) Lisp.

Prolog offers powerful data structuring capabilities (terms, lists, etc.) and supports specification of relationships in terms of two types of clauses: facts and rules, where a fact can be viewed as a special a rule with its if part missing. Indeed, a Prolog program consists of not much more than a set of such clauses. Figure 3 presents an example of a Prolog rule corresponding to the Rule 020 in Figure 2. (Note that in Prolog the comma represents the *logical and* operator.)

FIGURE 3. Prolog version of the Math Advisor Rule (Rule 020).

```
next_educ(e201) :-
        not taken(e201), standing(Year),
        member(Year, [sophomore, junior, senior]).
```

Prolog overcomes all of the drawbacks stated in the preceding section, and is general enough to be applicable to a much larger problem domain. On the other hand, Prolog does not provide automatically the WHY and HOW explanation capability that expert system shells provide. It does not provide certainty factors mechanism for dealing with uncertainty, nor does it provide a unified (restricted) rule expression system, or the automatic indexing of rules and their usage throughout the knowledge base. Most Prolog implementations provide access to other programming languages; in fact, the better Prolog implementations allow a Prolog program to be imbedded in a procedural language as well as to call in procedures written in such languages (C, Fortran, etc.).

Prolog is available in a number of language processor forms (compilers, interpreters). Commercial packages implementing Prolog are available for PC computers (e.g. Arity Prolog), as well as workstations (Quintus Prolog), and mainframes (MProlog).

10.0 Conclusions

An expert system shell, such as PC Easy, offers to the novice at expert systems a painless and protective manner to test the concept of prototyping. Much of the early mechanical parts of declarative programming are made easier by the Editor and implied structure that the expert system requires and the prompts and menus built in. You can add rules a little bit at a time, test them, and modify as needed; thus you get early feedback on the functionality of the prototype that you are building and whether it is behaving as needed or desired. For a novice, the logic of backward chaining becomes more real. Hence considerations as to good or better logic flow and writing of rules become more realistic and more heuristic. Debugging facilities ease the problems of logic checking and help you to understand backward chaining logic better. Most striking of all is the required reversal of thing of procedural programming when one doesn't have Go To's, structured Blocks, Do Whiles, Do Ends, etc.

The concept of prototyping gives practice in stating and determining good declarative rule construction. It teaches one to build logic decision trees in a declarative manner and the structuring of these so as to allow decisions in the same manner as does the human expert. It early suggests and rewards the process of seeking out and building of intermediate variables that clarify rule construction. An example of this is that in mathematics education finishing calculus is one major milestone and completing all of Math 212, 245, 255, and 295 is another major milestone. The prototype also allows one to pick a problem small enough to solve and on which to test your solution. With that experience behind you, analysis and design of the larger non prototype problem can then be investigated with prior knowledge of techniques, logic, and facts that have been thought out more completely and tested more fully.

11.0 References.

Bykat, A., Schmidt, R.J. *"A Diagnostic Expert System for Straight-Tube Welding Machine"*, Welding Journal, pp.40-43, (7)1989

Bykat, A. *"NICBES-2, a Nickel Cadmium Battery Expert System"*, Int.J.Appl.AI, pp.133-141, 4(3)1990

Bykat A. *"Intelligent Monitoring and Diagnostic Systems: a survey"*, Int.J.Appl.AI, pp. 339-352,5(4)1991

Waterman, D.A., "A Guide to Expert Systems", Addison-Wesley, 1986

Weiss, S.M., Kulikowski, C.A., "A practical Guide to Designing Expert Systems", Rowman & Allanheld, 1984

PAPER NO.: 6805

TITLE: Open Systems

AUTHOR: Birket Foster
M.B. Foster Associates Ltd.
P.O. Box 580
50 Water Street
Chesterville, Ontario K0C 1H0
CANADA
613-448-2333

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING, HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.

PAPER NO.: 7501 - 7506

TITLE: HP Strategy Sessions

AUTHOR: For more information please contact:

Ella Washington
Hewlett-Packard Co.
19091 Pruneridge Avenue, M/S 46LG
Cupertino, CA 95014
408-447-1053

HANDOUTS WILL BE PROVIDED AT TIME OF SESSION.