JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 41, No. 1, January–February 2004

Knowledge-Based Approach to Facilitate Engineering Design

Satwiksai Seshasai* and Amar Gupta†
Massachusetts Institute of Technology, Cambridge, Massachusetts 02140

A knowledge-based approach is presented to facilitate the engineering design process relating to spacecraft. The degree of collaboration across temporal and spatial boundaries plays a major role in determining the aggregate time and cost involved in each instance of spacecraft design. A major aspect of such collaboration is the issue of communications: the ability to capture the detailed needs of every stakeholder in the process, as well as rationale for the major design decisions. The approach described provides a framework for facilitating the decision making process in engineering design, by eliciting and capturing the goals and requirements of every stakeholder in the design process through utility and expense functions. An interactive system has been designed that incorporates a four-faceted knowledge-based framework of knowledge acquisition, knowledge discovery, knowledge management, and knowledge dissemination. We describe the combination of the multi-attribute interview software tool MIST and Space Systems Policy Architecture and Research Consortium (SSPARCy) paradigms to develop an evolving knowledge repository that enables one to perform crucial applications whose success is today contingent on geographical proximity. The proposed knowledge-based approach can be readily adopted to facilitate other applications that involve sustained collaboration across geographic and corporate boundaries.

Introduction

THE spacecraft design environment involves sustained effort by multiple teams and multiple stakeholders based at various locations. In such a distributed environment, it becomes especially important to be able to capture the needs clearly and efficiently of every stakeholder in the process, as well as the details of the major decisions and the rationale behind these decisions. By transferring relevant knowledge from one environment to another, one can provide major improvements to the design process and also enable leveraging of knowledge from one endeavor to another.

This paper proposes a multifaceted approach to provide a common interface to assist members of the design process to collaborate, to share goals, and to help formulate overall rationale. In addition to helping to create a knowledge repository for the specific design process on an incremental basis, the proposed approach facilitates knowledge discovery by presenting and processing data from all stages of the design process. Detailed material on the "why" aspect of spacecraft design is incorporated by capturing the true cost and utility to the customer.

The software tools described in this paper seek to solve the problem of collecting, sharing, using, and interpreting knowledge from various stakeholders in the spacecraft design process. We describe a concept demonstration prototype that incorporates methods used in other domains to provide infrastructure for the spacecraft design process. This paper will first describe the knowledge-based approach used as the overall framework, then it will describe both the design rational aspect and the utility interview tools, and finally, a test implementation of the tools will be described to provide a perspective on the unifying capabilities of our approach.

Four-Faceted Knowledge-Based Approach

In the specific context of design of spacecraft, knowledge-based techniques can be applied to achieve engineering goals more effectively by concurrently leveraging multiple facets. These facets are knowledge acquisition, knowledge management, knowledge discovery, and knowledge dissemination.[1] As a design framework, these facets ensure that an application includes capture of information, mitigation of heterogeneities in the underlying contexts of information, analysis of huge data sets, and extraction of relevant pieces of information.

Based on the preceding framework, tools have been developed specifically for the spacecraft design environment, the Space Systems Policy, Architecture, and Research Consortium tool (SSPARCy) and multi-attribute interview software tool (MIST).

SSPARCy is designed to perform automatic capture and processing of vital information as the design evolves over time. SSPARCy incorporates a suite of features that provide users with a centralized source of information regarding a design simulation; this centralized information repository allows the designer to analyze the evolution of a simulation over time. Furthermore, it performs an automatic analysis of system integration integrity to alert an integrator of potential problems.

The MIST system is an interface for capturing, processing, analyzing, and storing information about utility characteristics for every stakeholder in the design process. MIST provides a framework for expressing utility as described in the multi-attribute tradespace exploration (MATE) process.[2] MIST structures the knowledge and builds a series of rule-based interviews to elicit the stakeholder's utility. The MIST paradigm incorporates analysis tools that allow users to observe links between multiple stakeholders, multiple design phases, and multiple projects. Furthermore, emerging data mining techniques are being incorporated into the MIST system to provide knowledge discovery capabilities geared specifically to the spacecraft design community. The MIST system is currently designed to handle each stakeholder individually and can be readily extended to analyze data from multiple stakeholders. It is designed as a complement to SSPARCy.

Previous Efforts

Before we present details of our knowledge-based approach, we present information on related efforts by other researchers.

Vehicles Knowledge-Based Design Approach

The Vehicles Knowledge-Based Design environment[3] is a framework for managing knowledge related to the design environment of space systems. Analysis and modeling tools are combined with a historical database of previous projects to assist in building new architectures that benefit from the experiences of old architectures.

Received 16 January 2003; revision received 21 May 2003; accepted for publication 12 June 2003.

*Graduate Student, Electrical Engineering and Computer Science; currently Software Engineer, Team Collaboration, IBM, Westford, MA 01886.
†Codirector, Productivity from Information Technology Initiative.

The issues addressed in Vehicles are similar to the issues discussed in this paper and relate to the best methods for formalizing the knowledge related to the design environment. Vehicles proposes a flexible environment in which to describe systems and subsystems and to analyze the effects of various changes to the design. The software environment provides a platform for designers to build their own tools, tailored for their specific types of analysis. The design rationale, history, and utility capture capabilities described in this paper can provide a complementary value proposition to the Vehicles system, which was conceived in 1993.

Rule-Based Algorithms from Chung-Hua University

Another system[4] uses a rule-based algorithm for transferring an individual customer's needs directly into specifications, by developing a matrix of weights between attributes and design factors. The weights are then used to determine rules, or relations, indicating how certain design parameters should change their respective values based on other values of parameters. The paradigm proposed by us can build on this approach by incorporating information on the goals and needs of multiple stakeholders, as well as by using the evolving knowledge repository to facilitate decisions related to the derivation of specifications from customer preferences.

Distributed and Integrated Collaborative Engineering Environment

The distributed and integrated collaborative engineering environment (DICE) methodology offers a platform for collaborative engineering by decomposing each engineering project into a set of modules and allowing work to be conducted in parallel on each section of the project.[5] When the system encounters conflicting decisions about a particular design decision from engineers in different modules, it uses the design rationale to help negotiate the outcome. The approach described in this paper complements the DICE approach by providing a concrete relationship between the history of the design parameters and the associated utility and expense functions that led to these decisions. The use of these relationships would allow engineers with conflicting solutions to gauge the exact implications of each option on the different stakeholders of the system.

Utility Evaluation from University of Massachusetts

The tradeoff-based robust modeling and design group has developed an online utility evaluation tool that conducts interviews to determine stakeholder preferences in a similar fashion to the MIST approach.[6] The system incorporates mechanisms to deal with preference consistency, uncertainty, and risk. These issues are addressed in the context of a given stakeholder for a given project. The MIST approach builds on this methodology by capturing information that will be useful in determining the relations between preferences of multiple stakeholders, at multiple stages of the design process, and for multiple projects.

Knowledge-Based Module for Design Rationale Aspect

Consistent with our multifaceted knowledge-based approach, our research team has developed concept demonstration prototype tools to illustrate how such tools can incorporate knowledge that evolves over time.

Architecture for Design Rationale Concept

We first describe SSPARCy, which is a knowledge-based tool that can facilitate design and development tasks over time by ensuring that vital design decisions and rationale are appropriately encapsulated. By providing the user with detailed system analysis, history reviews, and error checking capabilities, SSPARCy attempts to address the void that currently exists for methods to capture crucial information automatically with no or little human intervention. The software imposes minimal overhead on the user. Once the project is set up, the software operates in the background, extracting the parameters values from the design environment, and importing these values into the SSPARCy data model. The users can then use SSPARCy to view and analyze these data and also provide additional inputs to SSPARCy on the design rationale, as deemed appropriate by the user.

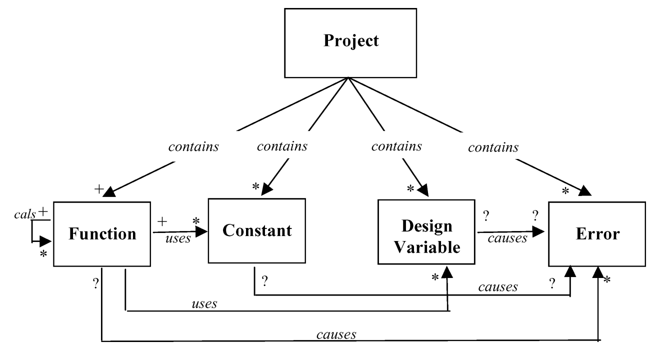

Fig. 1 SSPARCy data model; major system objects interacting over time.[7]

Each simulation exercise can be encapsulated into one major object, which is referred to as Project. The Project object contains all of the necessary objects and variables that represent the information stored in the application, such as functions, constants, design variables, and errors. The Function objects refer to actual functions in the simulation source code. Similarly, the Constant and Design Variable objects represent each global constant and design variable that is defined in the system. Last, the Error object refers to any possible error in the simulation design that can lead to redundant, unused, or misrepresented code in the code files. The architecture of the data model is shown in Fig. 1 (Ref. 7).

Each Function, Constant, Design Variable, and Error object can be referred to as a general Variable object. The architecture has been designed so that each Variable object can contain detailed information on the Variable's name, value, units, valid range, author, date of creation, and possible aliases that can refer to it in the project. Also, each Variable stores its respective rationale, so that the user can record design decisions and changes that relate to each object in the system. The data that are stored in the object model provide significant possibilities for greater functionality in the graphical user interface.

Knowledge Acquisition Aspect of SSPARCy

The graphical user interface has been designed to facilitate knowledge acquisition through intelligent capture and visualization of simulation exercises. SSPARCy facilitates project management efforts by allowing multiple projects, specified in different formats, to be opened, viewed, and saved simultaneously. The system also enables the user to view any Variable object in the current project and to see all of the vital information fields that are stored along with that Variable. As the system design evolves over time, variables may be added, removed, or changed from the current status. Finally, the current implementation incorporates functionality for error checking so that users can see what possible errors might exist in the project and where those errors might have occurred.

Knowledge Management Aspect of SSPARCy

Current Variable Data

Whether the user wants to see the value and the rationale of a global constant in the project or just the name of the author of a specific design variable, SSPARCy provides multiple display options in the form of tables that display the changing state of the data.

Variable History Review

For every Variable object in the system, the user is able to review quickly how that variable has changed over time and what variables have been added or removed from the current project. The use of the colored format is just one representation of the concept that history of the variable in the grid should reveal evolving trends in a visual manner. By allowing the user to view the history of any

Variable object in the system, SSPARCy provides an extensible tool for comprehensive analysis of successive simulation exercises.

Knowledge Discovery Aspect of SSPARCy

To ensure that the design rationale capture aspect imposes zero or minimal overhead on the designers, SSPARCy parses parameters' values, units, comments, and timestamps from their source files. However, if designers wish to enter additional details regarding a parameter, they can do so by selecting the parameter from the table and then clicking the button Edit Info.

To support better inspection and analysis of future simulation files, an approach was developed to provide the following functionality: Permit the viewing and storage of important information relating to the functions, global constants, and design variable that are used in the project, enable storage of the history of the data as it changes over time to enable the user to know exactly what information has been updated since the last system design, and provide for error checking of the current system design to inform the user as to where possible errors may exist in the simulation and modeling files and how those errors may be fixed.

MIST

MATE and its follow-on concurrent design analog MATE-CON were designed to facilitate intelligent interaction with the designer and customer. This process adapts the multi-attribute utility analysis (MAUA) method[8] to the space system design domain. MAUA provides a technique for capturing the value in terms of utility, which is determined by preferences for attributes and risk. The MAUA method introduces a mathematical representation for users with multiple objectives. By using this representation, we seek to improve the space system design process, a process in which many stakeholders with many complex objectives have ultimately to agree on one design.

Instead of relying on a cumbersome face-to-face interview process, MIST uses an advanced graphical user interface and a Web-based computer interface with graphics to speed up and enrich the utility interview process. One goal of MIST is to demonstrate the value of software for conducting dynamic questioning sessions that gradually lead to the creation of the utility function for the particular application domain. Because the space system design process is very complex and involves many changing variables, the availability of a methodology for continuously assessing the utility of major stakeholders will greatly help to keep the design process focused on what will ultimately become the best design. In this software system, as shown in Fig. 2, the designer is able to describe attributes and their respective ranges of values, units, and scenarios, based on input from the customer. For example, in a satellite design project, attributes could include data life span, time spent in orbit, and sample altitude. The system then prepares an interview based on the attributes of the tradespace and allows the designer to conduct the interview and to enter the responses. Furthermore, the customer can take the interview independent of the utility facilitator. The MATE interview process consists of multiple stages, dealing first with single attribute utility parameters and then with multiple attribute utility parameters. Each stage also contains a set of validation questions to ensure that the variables being considered are independent of each other.

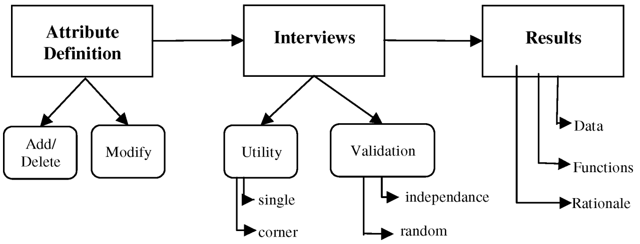

Fig. 2 Overview of major steps that MIST can automate.

The automated and customized interview sessions present the customer with a scenario using the lottery equivalent probability approach as developed in the field of decision theory.[9] Each option is a situation characterized by a set of probabilities of two states for purposes of attribute analysis. The user will usually express preference for one of these two states. In response, the system will modify one of the states and ask the question again in a manner very similar to the series of dual-value choices given by an optometrist when fine tuning the power of the lens to correct the eyesight of a patient. The system continues in this form until the customer indicates indifference to the two options. The software then tallies the indifference points for later calculation and moves on to analyze the next attribute. This entire process can be conducted on a single day or over time. The results of each interview are utilized for developing functions to assess the utility and the cost of design, based on the set of the attributes specified by the customer or the set of customers. The integration of these data with the design parameter data collected by the SSPARCy system provides a comprehensive assessment of the rationale at various levels of design detail, throughout the design process.

Knowledge Acquisition Aspect of MIST

By conducting the interview with a software tool, the process can be made fully interactive and tailored to the individual responses, through dynamic calculations of utility. Based on the input from the designer and the stakeholder on the process, a repository is created that incorporates both the design of the system and the information gained during the interview process. The major issue raised in this area is how much control and flexibility is appropriate in the customization of the interview session on a real-time basis. Allowing excessive variability in the interview process may reduce the reliability and consistency of the results.

Knowledge Management Aspect of MIST

In the concept demonstration prototype endeavor, the paradigm of customized visualization adds value to customers by helping them to understand the way in which their utility estimates graphically evolve over time, with both positive and negative impact. Based on the problem domain, the designers can specify the amount of dynamic visualization allowed, thereby controlling the amount of knowledge the stakeholders can use to "game" the system to produce the utility functions they particularly desire.

Knowledge Discovery Aspect of MIST

One of the major intellectual questions that MIST attempts to address is the interdependency of utility- and attribute-based design rationale. When one integrated system is used to collect both forms of data, one possible strategy is to relate each moment of the design process to a specific interview, thereby associating an interview with each value of a design parameter.

Knowledge Dissemination Aspect of MIST

The MIST system contains a set of analysis tools to process the data collected from the interviews and the SSPARCy design parameter capture process. The use of an integrated tool allows for a consistent template to be used, as well as for a comprehensive knowledge repository to evolve over time. In addition to producing the reports, algorithms for utility and expense function generation are integrated with the system to produce relevant functions that incorporate the information from the interview process. Customized utility functions for each stakeholder allow all iterations of an architecture to be assessed, and the changing utility of an architecture to be related to the evolving preferences of the customer.

Attribute Interface

The MIST system utilizes the notion of attributes to represent the user-defined characteristics of the project that describe the important factors for the stakeholder. Attributes are defined through a series of discussions between the different members of the engineering team, as well as the stakeholder. The attribute interface has been designed to structure the important pieces of knowledge related to each attribute, so that the discussion can be focused. Attributes in the system must be defined before the interview session begins, so that all interviews are conducted within the same context.

An attribute is represented in the system as a hybrid data type with various properties, as shown in Table 1. Table 1 shows the properties related to an attribute, the data type of each property, the description, and comments. Some of these properties are directly related to the description of the system, and others were implemented to facilitate the software system. When the design team creates an attribute, an instance of the attribute data type is instantiated and modified as appropriate. The interview interfaces and the analysis tools base their actions on the status of the attribute properties. An example of a fully defined set of attributes for a system, taken from the SSPARC group's terrestrial observer swarm iteration B (B-TOS) project, is shown in Table 2 (Ref. 8).

A user can add, modify, or delete an attribute and its associated properties, as shown in Fig. 3. Attributes are defined for each stakeholder, through a discussion with the designers of the system. This discussion aspect is perhaps the most important phase of the MATE process. When the handful of most relevant properties are defined, the stakeholders are actually defining the factors that are most important to them in the design of the system. As such, the rationale behind major decisions made during this process is elicited and stored along with the attribute properties. Stakeholders are prompted to specify whether changes to an attribute's properties constitute a major change and, if so, are prompted for the corresponding rationale. Even attributes that are considered and then deleted are still stored within the system, to preserve the full history of the decision making process.

Table 1 Properties for attribute data types in MIST

Attribute property	Data type	Description	Comment
Name	String	Name of attribute	Understandable to all involved
Min	Double	Lowest value in range	
Max	Double	Highest value in range	
Units	String	Units for attribute values	Must be very specific
Increment	Integer	Number of indifference points to be collected	Enough to avoid large jumps in utility
Direction	Boolean	Direction of increasing utility	Toward max or min value?
Scenario	String	Scenario for single attribute interview	Describes context in which only the single attribute is considered
Resolution	Double	Probability resolution to which utility can be distinguished	Usually around 5%
Format	String	Format in which attribute is presented to user	Examples: 5 × 5, 6
UnitsInc	Boolean	Include units in format?	Usually true
Independent	Double	Attribute value for independence interviews	Usually near middle of attribute range
Definition	String	Definition of attribute	Must be very specific
LinearScale	Boolean	Increase indifference point questions on linear or logarithmic scale?	
Threshold	Double	Difference in utility between adjacent indifference points which requires more questions	

Characteristics of Interview Process

Each interview is designed to acquire a piece of knowledge intended to help build or validate a utility function for the stakeholder being interviewed. The interviews are meant to take place after the attribute definition phase has occurred and after the stakeholder has approved the attributes. The interviews can be conducted at multiple points throughout the design process, to provide a continuing notion of the important goals and requirements of the stakeholder being interviewed.

Visualization

The issue of visualization is one that was examined in great detail by the designers of the MIST system. When the MIST concept demonstration software is used, it is possible to display the curves representing the utility functions for the attributes being discussed as the interview occurs. Figure 4 shows this functionality for the single attribute interview, where the responses from the user gradually help to create the utility function for a single attribute. Similarly, the utility curves generated for all of the attributes gathered in any of the multi-attribute interview sessions can be displayed.

If a response during the single attribute interview leads to a drastic change to the utility curve, the interviewee can see this immediately and think again about whether the response accurately represents preferences. During the multi-attribute interviews, when the user is answering questions dealing with the relationships between attributes, the availability of displays of utility curves greatly facilitates the ability of the interviewee to evaluate actual feelings about the different attributes.

When this idea was presented to the fellow designers, the decision was made to withhold the visualization features from the stakeholder because it might provide excessive information and consequently lead to bias in the responses. For example, during the latter part of a session, a stakeholder might be inclined to provide responses that fit the initially defined utility curve, to avoid the appearance of being irrational. Another possible downfall of the visualization feature is the ability of the user to predetermine a utility curve in their mind and to provide answers that fit this curve. Based on further tests, one will decide whether to offer the extended visualization capability as an option in future versions of MIST.

Table 2 Sample attribute definition from B-TOS project[8]

Attribute	Definition	Best	Worst	k
Spatial resolution, deg	Area between which you can distinguish two data sets	1 × 1	50 × 50	0.15
Revisit time, min	How often a data set is measured for a fixed point	5	720	0.35
Latency, min	Time for data to get to user	1	120	0.40
Angle of arrival (AOA) accuracy, deg	Error of AOA measurement	0.0005	0.5	0.90
Electron density profile (EDP) accuracy, %	Error of EDP measurement	100	70	0.15
Instantaneous global coverage, %	Percentage of globe where measurements are taken in a time resolution period	100	5	0.05
Mission completeness	Mission type conducted	EDP/AOA and turbulence	EDP only	0.95

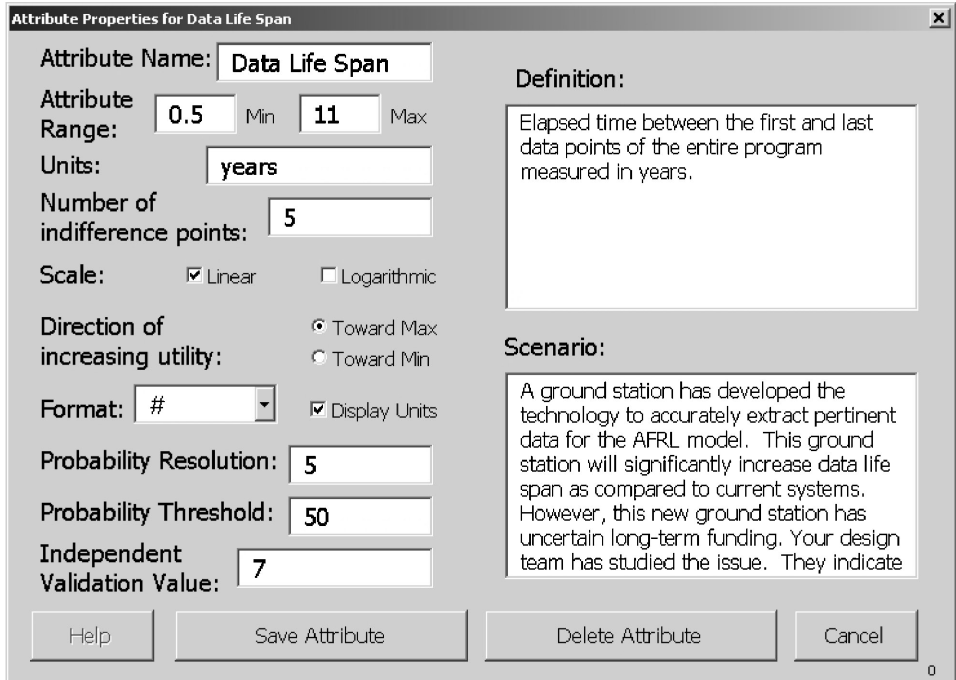

Fig. 3 Attribute modification interface.

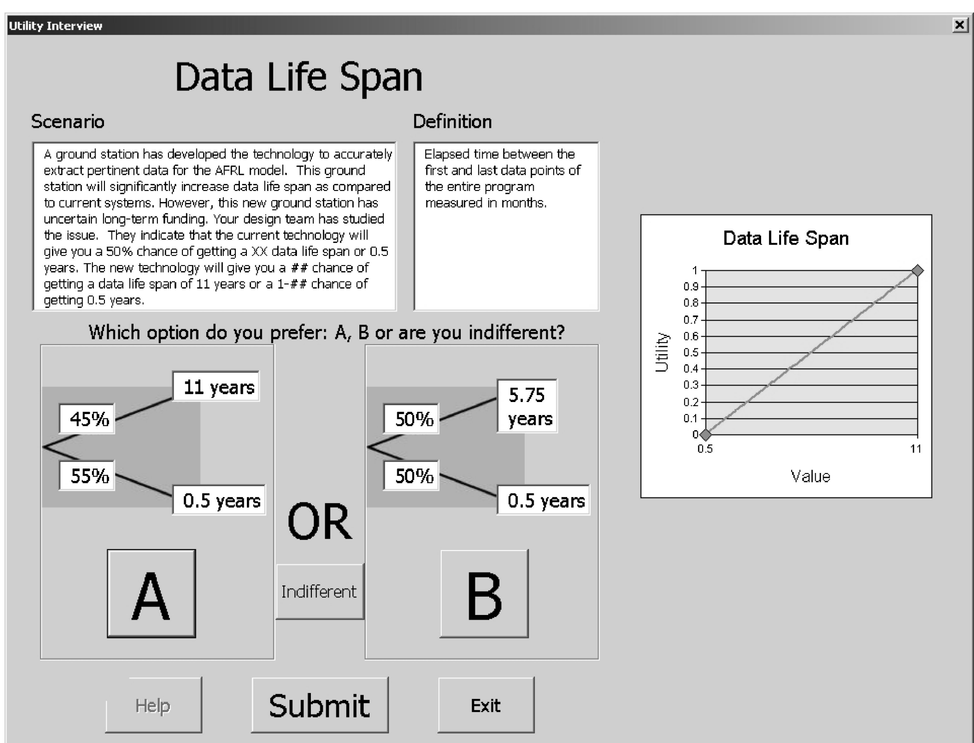

Fig. 4 Single attribute interview with utility curve.

Bracketing

Every interview generated by this system is geared to find the value at which the interviewee is indifferent between the two situations presented. The decision is basically between one option where the probabilities of certain outcomes are fixed, and a second situation where the probability of one outcome, as opposed to another outcome, is varied. This is based on the principle that users will seek to maximize their expected utility, and this will reveal the perceived risk and uncertainty.[9] The interview paradigm is designed so that the probability varies between 0 and 50%, a probability greater than 50% would result in one outcome always being better than the other for the user.

The specific method for obtaining successive values is based on the algorithm used in Ref. 10. At any given time, there is a known range of probabilities in which the indifference point exists. For example, when an interview question begins, it is known that the indifference point is within the range of 0% to 50%. The software module in the MIST system that

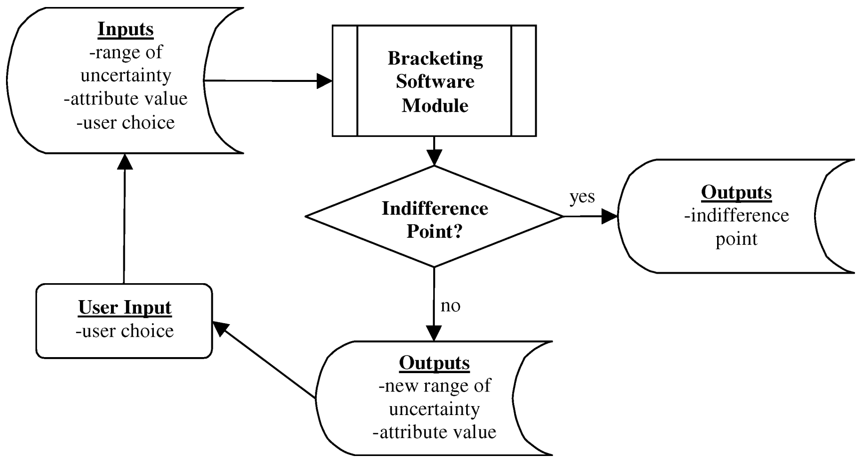

Fig. 5 Flowchart of bracketing software module.

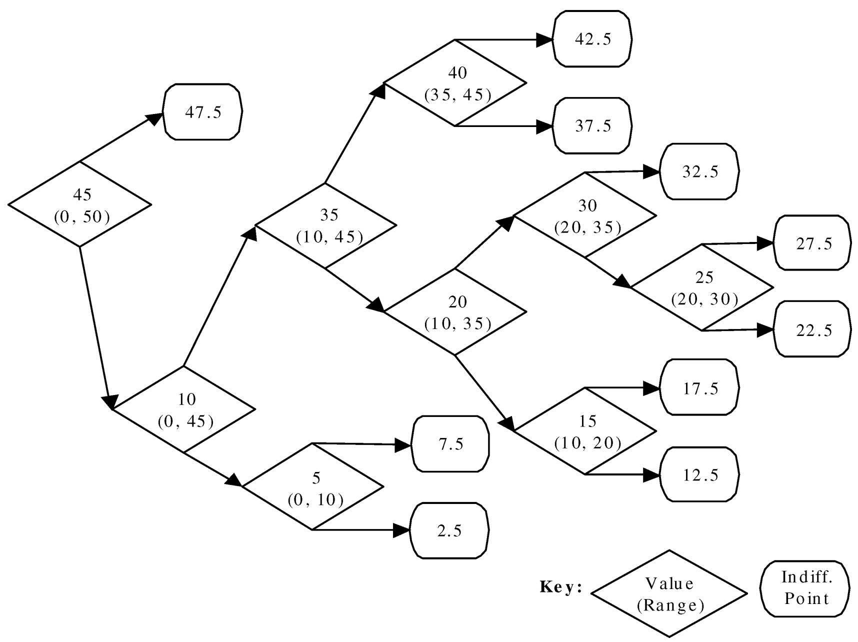

Fig. 6 X-TOS bracketing decision tree.

generates potential indifference points is shown in the flow chart in Fig. 5.

Each successive question asks the user for a preference at the midpoint of this range. Based on the user's response, it will be clear whether the indifference point is either equal to, above, or below the probability specified in the particular interview question. Either the interview ends with an indifference point being selected, or the range is modified to reflect the latest response and a new question is asked. Each probability chosen is a multiple of the attribute's probability resolution, which represents the degree to which the customer can distinguish between the two situations. This process continues until the stakeholder declares a certain probability to be the indifference point, or alternatively, the size of the range is less than the specified resolution. At this point, the stakeholder is informed that the indifference point will be set to the value exactly between the two endpoints of the range. A modification to this bracketing procedure was made for the X-TOS implementation described in this paper, as described in the "Test Implementation" section. Figure 6 shows the example from the X-TOS project. The probability resolution for this example was 5%; instead of using the midpoint of the range, values were chosen closer to the endpoints. Thus, the first node on the tree, the initial question posed in the interview, is for 45%. At each node in the tree the interviewee's choice indicates that the indifference point is either higher or lower than the current node's value. This choice results in a modified range of possible indifference values and a new node. For example, in Fig. 6, if the interviewee indicates that the indifference point is higher than 45%, then the indifference point is set to 47.5% because the new range is equal to the probability resolution of 5%. However, if the interviewee indicates that the indifference point is lower than 45%, then the next node to be reached is 10%, and the range is modified to (0, 45). Note that, although the most efficient means of conducting a binary search to reach the indifference point would appear to be continuously asking about the midpoint of the range, it is easier for the interviewee to think about their preferences if they are eased into finding the indifference point by starting at the edges of the range.

Single Attribute Interview

The goal of this interview is to build a utility function for each attribute. The MATE process adopted the single attribute utility function method from von Neumann–Morgenstern (see Ref. 11). This function specifies the mechanism for assessing the utility corresponding to differing values of the attribute for any proposed architecture and provides decision makers with knowledge about whether one can obtain a significant gain in the overall utility function by increasing values, either at the lower end of the attribute range or at the high end of that range. The utility functions provide both a visual notional idea of the nature of the attribute, as well as a concrete input into a concurrent engineering simulation model. A concurrent engineering simulation can use the utility function, combined with a module that calculates the values of each of the attributes based on the architecture, to assess the utility provided for each attribute in any proposed architecture. This offers the potential to provide real-time feedback to engineers about a potential design change, without having to consult the various stakeholders every time.

After the attributes are defined and the properties are agreed on, the system is given back to the stakeholders with whose assistance the attributes were designed. Depending on the nature of the roles involved, a single set of attributes may apply to a single stakeholder or a set of stakeholders with similar roles. The stakeholder begins an interview session with the single attribute interview.

The stakeholder navigates through the single attribute interview, as shown in Fig. 6, for each attribute. The user is provided with two pieces of information: the scenario and the definition of the attribute. The scenario is written to place the question within the specific context that emphasizes that the specific attribute in the interview is the only aspect of the system to be considered at that time. Thus, a situation can be described, such as the discovery of a new technology that has the potential to affect the value of the attribute, but carries with it some risk of failure. Identifying the level of risk that the user is willing to take is one of the purposes of the single attribute interview. The definition of the attribute is also displayed because, in many engineering situations, the specific interpretation of a concept may vary among different parties.

The interview begins with a random value extracted from the list of values generated with the Attribute Property form. The bracketing method described earlier is used until an indifference point is reached. Once all of the indifference points are collected, the system uses the Utility Threshold collected by the Attribute Properties form to determine whether new indifference points need to be collected. When this process is complete, the single attribute utility interview ends, and a check mark is placed in the appropriate cell on the main page. The user is returned to the Attribute Navigator form, and can participate in further interviews for remaining attributes, as appropriate. Figure 7 shows this process flow.

Corner Point Interview

The goal of the corner point interview is to determine the relative importance of each attribute with respect to the other attributes in the system by finding the corner points as described in the MATE process.[8] This is done for each attribute by presenting the stakeholder with a situation similar to the one shown in Fig. 8. The user interface used for the corner point interview is similar to the interface for the single attribute interview, except that individual values are replaced with lists that contain values for each attribute.

Furthermore, the certainty equivalent method is used, in place of the lottery equivalent method.[8] In the certainty equivalent method, the user is presented with a choice between being certain of a particular outcome and having a chance of another outcome. In the corner point interview, the two choices are as shown in Fig. 9: The different method is used because the goal of this interview is to determine the relative weight of one attribute vs the others. In Fig. 8, choice A represents the certainty that all attributes will be at their lowest level, except the given attribute, which will be at its highest level. Choice B represents a 90% probability that all attributes will be at the ideal value, and a 10% probability that all attributes will be at the lowest value. This probability is varied in an iterative manner using the bracketing methodology described earlier, until the user reaches the indifference point between two given choices.

At this point, the probability value, or k value, is stored with the attribute's parameter data. The k value represents the relative weight, from 0 to 1, of the specific attribute being interviewed. A higher indifference point in a corner point interview means that the stakeholder was willing to give up a chance at having a perfect system at a specific probability in exchange for the certainty of having a system where only the given attribute is perfect. The higher this probability, the more important the attribute, and the higher is the k value.

Attribute Independence Interview

After the data are collected, one begins to validate the selected attributes by ensuring that they are independent of each other. This is done by assessing and comparing the utility of two parallel architectures. For each of these two architectures, the value for the attribute being tested is left constant; its value is the one specified by the users in the attribute definition stage early in the process. The difference between the two architectures is that the remaining attributes are all set to the highest values in one and the lowest values in the other. The stakeholder is asked to consider each system's utility with respect only to the attribute in question. If the attribute is truly independent, the utility of both architectures should be the same.

Random Mix Interview

The Random Mix module was developed to provide data to validate the utility functions developed by the MIST system. Users of the system can specify the number of random sets to generate. For each set, the system employs a random number generator that selects a value for each attribute along its range, as specified in the attribute

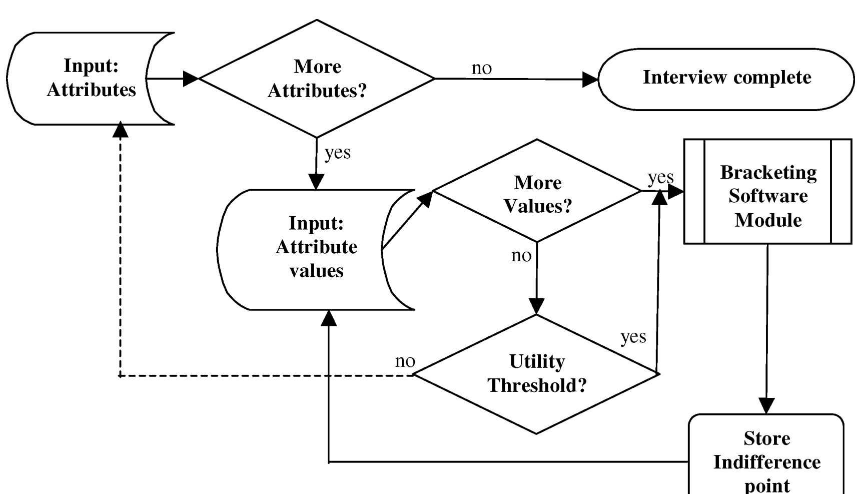

Fig. 7 Process for single attribute interview.

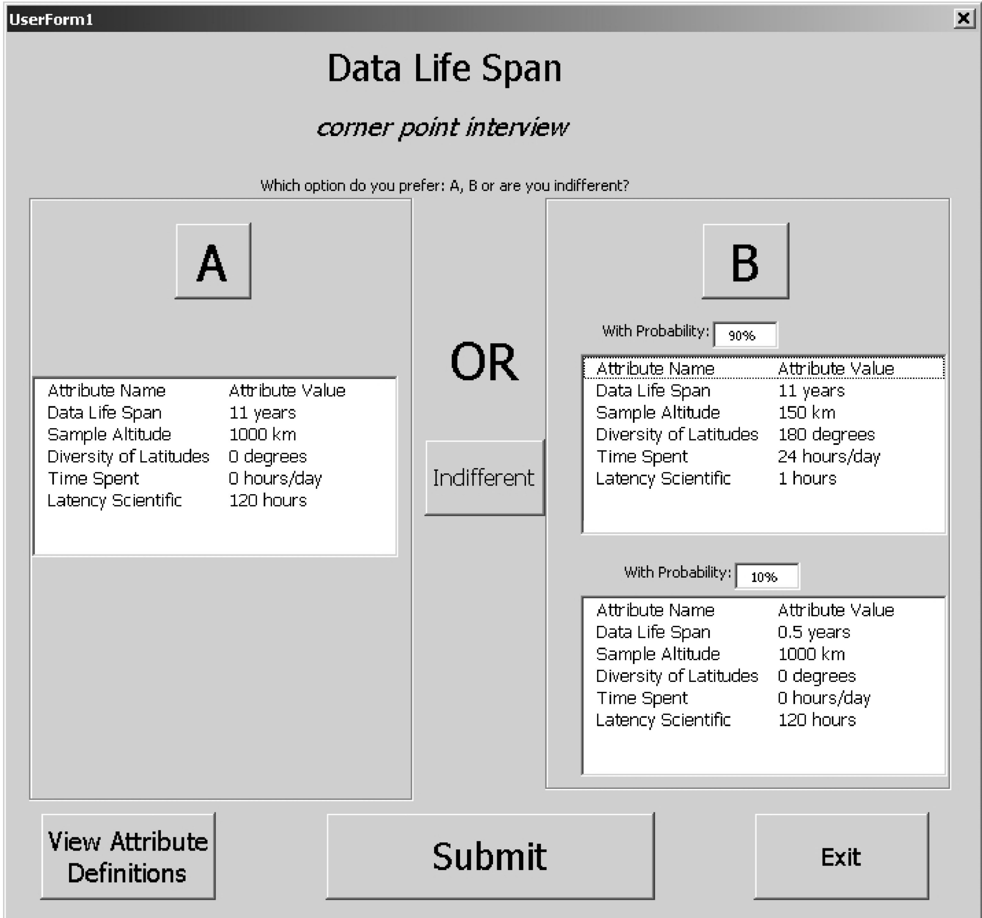

Fig. 8 Multiattribute interview form for corner point and random mix interviews.

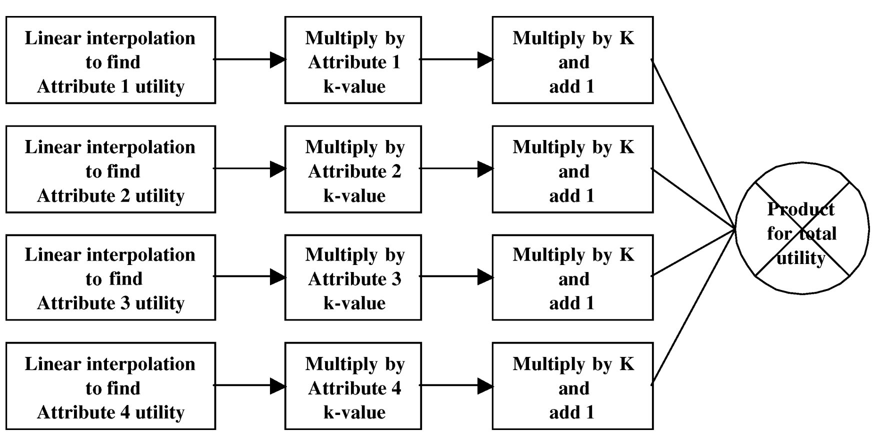

Fig. 9 Multiattribute utility calculation: individual utility functions for each attribute combined to produce a utility function for the overall system.

definition stage. The resulting random sets, each representing a random architecture in the tradespace defined by the stakeholder and designers, are presented to the stakeholder as the culminating set of questions for the interview session. With use of the same interface as the corner point interview, the system uses the bracketing module to determine the utility of each of the random sets. These utilities are compared to the utilities generated by calculating the utility of each of the architectures studied with the utility functions. Often, these utilities do not correlate, leading to the conclusion that humans cannot properly comprehend the multidimensional problem of determining their own utility. Accordingly, a system such as MIST helps to distill the stakeholder's true utility values[8] more accurately.

Output of MIST

The MIST system produces a set of outputs that can be used to facilitate the knowledge discovery and knowledge dissemination facets described earlier. The outputs described here are for one session of the MIST interview process.

Attributes and Interview Reports

Data in the MIST system are stored with respect to each attribute. As the interview is conducted, the responses are stored in the same spreadsheet that contains the attribute's properties as conceived in the attribute definition stage. Every choice is recorded; however, only the indifference points are used to calculate the utility functions and k values. At any time, the user can opt to have all indifference points collected and stored in a separate table for analysis.

Single Attribute Utility Function

The core output for the MIST interview system is the utility functions for each attribute. As described in the MATE process, these utility functions describe the changing utility of the concerned architecture as the value of the attribute changes. When the MIST user selects the Generate Reports button, it is possible to generate a utility function for any attribute. The utility functions are represented as curves with each indifference point plotted and linear interpolation used to determine the function between indifference points.

Multi-Attribute Utility Function

The final output for the MIST system is the multi-attribute utility function, which calculates the utility of an architecture based on values that were established for each of the attributes. The derivation of the actual function is described in the MATE process and the equation is

$$KU(\boldsymbol{X}) + 1 = \prod_{i=1}^{n=6}[Kk_i U_i(X_i) + 1] \tag{1}$$

In the preceding equation, k_i and $U_i(X_i)$ represent the k value and utility function of a single attribute, and K is the aggregate scalar calculated as described in the MATE process. The equation provides a means of calculating $U(\boldsymbol{X})$, which is the overall utility function, where X is the set of values X_i that represent the attribute values of a given system X.

In MIST, code modules perform each step of the derivation on the fly, as shown in the flowchart in Fig. 9. Basically, once the value of each attribute is known, the utility of the system for each attribute can be determined by linear interpolation of the values at the two closest indifference points. Each attribute's utility is then multiplied by the k value, and then a normalized product is produced; this represents the utility for the system with the given set of attribute values.

Test Implementation

Project Description

The approach described in this paper was tested during the spring of 2002 as part of the SSPARC's X-TOS project. The project was conducted with support and inputs of members of The Aerospace Corporation and U.S. Air Force Research Laboratory.

In previous years, members of the design team assigned to assess the utility of the proposed system conducted manual face-to-face interviews with the customer. During 2002, the gathering of utility data was accomplished using the prototype software system. The role of the scientific customer in the X-TOS project was played by Kevin Ray, a member of the U.S. Air Force Research Laboratory at Hanscom Air Force Base. Subsequently, in an e-mail dated 25 June 2002, Ray stated:

> The software cut my interview time by over 50%. The process was straightforward and I could go at my own pace. In addition, I was asked more questions than last year and the software calculated the Multi Attribute Corner Points, Independence High and Independence Low and Random interviews for every attribute. The software was able to graphically show my preferences for each attribute and identify inconsistencies in one of my attributes. This software is a great benefit to customers because it shows them how prioritizing one attribute of another attribute drives the design of the system.

The software system also allowed this interview to be conducted without requiring all of the interacting parties be present in the same geographical location. In fact, the interview session was conducted in stages, as it fit in with the user's personal schedule, and the communication between the user and engineers during the interview session was made via telephone and electronic mail. The user was also able to take time off to consult with other members of the customer team because work was ongoing at the user's own schedule and own location. The flexibility in terms of both time and location contributed to higher quality of responses by the user. In previous projects, the burden of conducting the interview led to the utility information being collected from one stakeholder only and that, too, in a piecemeal manner. With the implementation of the approach described in this paper, the utility data can now be collected with a higher fidelity, with more stakeholders, and at more frequent intervals throughout the design process.

By increasing the frequency of entry of utility data, and by processing of successive streams of data, one can accomplish significant levels of knowledge discovery. This approach has also allowed our design team to engage in knowledge management and knowledge dissemination endeavors because the utility information becomes a continuous driver for the design process rather than being a simple one-time input into the system.

Issues Encountered

After using the system, Ray observed that the system was very intuitive and the interview process was much more understandable. Our interactions with him and other test users have led to several design refinements in SSPARCy and MATE, and the major issues are mentioned here.

The independent evaluators highlighted the importance of the attribute definition stage and the need for the SSPARCy and MIST paradigms to be better applied to this stage. The knowledge contained within the discussions leading to the final attributes is valuable and should be preserved. Another change necessitated by the X-TOS implementation was the incorporation of an interview override mode, which raised the issue of the optimal balance between manual inputs vs automation. This implementation also underscored the importance of a clearer and networked interface that would enable the designers to better communicate with the user before and during the process. As the system evolves, it will incorporate the ability to observe the state of the interview over the network. Finally, a mechanism for scheduling the various steps in the interview process would reduce the amount of familiarity the user must have about the process itself and make the system more automated and flexible.

Lessons Learned

The issues raised highlight that the effectiveness of knowledge acquisition is heavily dependent on the nature of the software interface. In designing such a system, it is important to consider the type of knowledge being sought, the form in which it exists before using the system, and the form in which it should be disseminated. The knowledge being acquired in this system exists at multiple levels. If users of the system are asked the question that our system is trying to answer, such as what is the relative importance of an attribute, they might give a certain set of answers. However, when the interview questions are framed in the certainty equivalent method employed by this system, the user may give answers that are closer to their true preferences. One needs to be able to transcend easily between these two levels. The X-TOS implementation process demonstrated that this goal is difficult to achieve. Apart from facilitating communication across users and locations, one needs to be able to elicit knowledge that the users may not even be consciously aware of.

Conclusions

Design environments such as the spacecraft environment are dependant on the successful interaction of multiple teams and multiple stakeholders based in various locations. Through the capture of the needs of every stakeholder in the process, as well as the details of the major decisions and rationale, one can support effective knowledge transfer between teams without requiring intense face-to-face interaction. Such a knowledge-based framework can have a tremendous impact on the design process, by providing designers with all relevant sets of information. The knowledge acquisition process was greatly facilitated by use of the interview tool that allowed for faster and more frequent generation of the utility function. The data collected by this system help the knowledge discovery process by exposing recurring issues and decisions. Our analysis tools provide new methods for knowledge management in the design process. Finally, knowledge dissemination to new organizations that enter the design environment is improved because the history is stored in a consistent and managed structure. The model presented in this paper provides a means for innovating the spacecraft design process by exploiting the underlying knowledge within a traditional design environment.

Acknowledgments

We thank fellow members at the Space Systems, Policy and Architecture Research Consortium, especially Daniel Hastings, Hugh McManus, Joyce Warmkessel, Adam Ross, Nathan Diller, and Quincy Scott. Additionally, members of the SSPARC IT team who assisted in the development of the software described in this paper are Alexandro Artola, Presley Cannady, Winston Chang, Shane Cruz, Anna Konfisakhar, Carolyn Ng, Frank Reyes, Tara Sainath, Connie Tao, and Jason Yeung.

References

[1]Gupta, A., "A Four-Faceted Knowledge-Based Approach for Surmounting Borders," *Journal of Knowledge Management*, Vol. 5, No. 4, 2001, pp. 291–299.

[2]Diller, N., "Utilizing Multiple Attribute Tradespace Exploration with Concurrent Design for Creating Aerospace Systems Requirements," M.S. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, June 2002.

[3]Gillam, A., "Vehicles Knowledge-Based Design Environment," *Journal of Spacecraft and Rockets*, Vol. 30, No. 3, 1993, pp. 342–347.

[4]Wei, C.-C., Liu, P.-H., and Chen, C.-B., "An Automated System for Product Specification and Design," *Assembly Automation*, Vol. 20, No. 3, 2000, pp. 225–232.

[5]Sriram, R. D., *Distributed and Integrated Collaborative Engineering Environment*, Sarven, Glenwood, MD, 2002.

[6]Jie, W., and Krishnamurty, S., "Comparison Based Decision Making in Engineering Design," American Society of Mechanical Engineers Design Theory and Methodology, 1999.

[7]Scott, Q. R., "SSPARCy: A Software Integration Support and Design Rationale Capture System," M.S. Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Inst. of Technology, Cambridge, MA, July 2001.

[8]Diller, N., Dong, Q., Joppin, C., Kassin-Deardorff, S., Kimbrel, S., Kirk, D., McVey, M., Peck, B., Ross, A., and Wood, B., "B-TOS: Terrestrial Observer Swarm," Final Rept., 16.89, Space Systems Engineering, Massachusetts Inst. of Technology, Cambridge, MA, Spring 2001.

[9]De Neufville, R., *Applied Systems Analysis: Engineering Planning and Technology Management*, McGraw–Hill, New York, 1990.

[10]Casey, J. T., and Delquié, P., "Stated vs. Implicit Willingness to Pay Under Risk," *Organizational Behavior: Human Decision Processes*, Vol. 61, 1995, pp. 123–137.

[11]Keeney, R. L., and Raiffa, H., *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Wiley, New York, 1976.

J. Korte
Guest Editor