CASE-BASED COST ESTIMATION

A BUILDING BLOCK FOR PRODUCT COST MANAGEMENT AND DESIGN-FOR-X

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen op
woensdag 24 september 2003 te 16:45 uur

door

Alexander Layer
geboren op 18 juni 1973
te Schorndorf, Duitsland

Dit proefschrift is goedgekeurd door de promotoren:

prof. dr. ir. F.J.A.M. van Houten en

prof. dr. ir. H.J.J. Kals

CASE-BASED COST ESTIMATION

A BUILDING BLOCK FOR PRODUCT COST MANAGEMENT AND DESIGN-FOR-X

Alexander Layer

Case-based Cost Estimation
A Building Block for Product Cost Management and Design-for-X

Ph.D. Thesis

ISBN 90-365-1961-6
ISSN 1386-5307

To my parents

"It's hard to make predictions – especially those about the future." Allan Lamport (1904-1999)

Preface

This thesis represents the result of my post-graduate research work in the field of cost estimation. It was initiated by Dr. Siegmar Haasis. I was able take up his ideas and mature them as a Ph.D. student in the Research and Technology Division of the DaimlerChrysler Group. The dissertation topic is related to several research projects undertaken by the department *Product, Process, Resource Integration* (RIC/EP) within the Lab *IT for Engineering* (RIC/E). In particular, two research projects were closely coupled with cost estimation: feature-based cost estimation and DesMIT (Design/Manufacturing Integration Technology). These two projects primarily aimed at bringing together the product description and the product costs and helped me to refine my ideas and to validate some of the results.

The research work at hand would not have been possible without Dr.-Ing. Siegmar Haasis. I particularly wish to express my gratitude to him for putting his confidence in me and for providing me the opportunity to accomplish this research work. He established the environment and the necessary degree of freedom for me to push the dissertation project. He was most helpful in offering his constructive criticism during our discussions.

Not less gratitude is owing Prof. Dr. Ir. Fred van Houten, who supervised the work on part of the university with a great amount of interest. His expertise and analytical way of thinking have always helped me to critically question the things I have elaborated. Despite of the distance between Ulm and Enschede, the discussions he was kind enough to grant me were sufficiently frequent to find their way in my research. In addition, I wish to thank Prof. Dr. Ir. Hubert Kals for assisting in guiding this research project on part of the university. Furthermore, I would like to express my gratitude and appreciation to Prof. Ir. Henk Bikker, Prof. Dr.-Ing. Helmut Bley, Prof. Ir. Arthur Eger, and Prof. Dr. Ir. Marc Wouters for serving on my committee. And I cannot neglect the researchers of *Laboratory of Design, Production and Management* at the University of Twente for providing me with the opportunity of detailed insight into their research work and for the prosperous discussions we had.

Without the cooperative attitude and the lively interest on the part of DaimlerChrysler, this thesis would not have come about. Simultaneously, this demonstrates the pivotal importance of cost estimation. The thought-provoking exchanges and fascinating discussions I had with the colleagues in my department and in the projects – even if sometimes hard fought – have helped me to accomplish this thesis: our collaborative efforts have borne fruit in this work. Each and every one of them has contributed valuable suggestions and advice that have assisted me in completing my work. Thank you all.

I would also like to mention the students doing practical internships and diploma theses related to the cost estimation research work and additionally, Larissa Glaser for proof-reading my thesis.

Finally, I owe more than my gratitude to my parents, Dieter and Ursula Layer, for supporting me above and beyond what I expected or could ever have hoped for.

Schorndorf, August 2003

Alexander Layer

Summary

To design desirable products with an increasingly high level of functionality and quality in an ever decreasing time-span is a demanding goal. In particular if the product's profitability has to be ensured simultaneously. Hence, enterprises today are faced with the necessity to cut product costs by, in particular, developing products which will incur lower production costs. Thus, fulfillment of functional requirements and assuring high quality are prerequisites for successful products; yet they are not sufficient on their own. Since the achievable profit is usually related to the product costs, it is insufficient to do accounting in retrospect: instead, there is a need to proactively analyze and continuously control product costs. Product cost management is the framework for all analysis and control activities to proactively influence and reduce product costs. But, as there is no appropriate tool for design-concurrent cost estimation, an adequate methodology is needed to sufficiently support the design engineer in developing cost-optimized products.

This thesis proposes a model for cost estimation of discrete mechanical parts in detail design. This model serves as a building block for product cost management and design-for-X. The thesis is divided into three parts:

- The framework.
- The cost estimation concept.
- The application of CABACO (case-based cost estimation).

In the first part, the product creation environment and knowledge processing are analyzed. Therefore, a short overview of the product creation process, feature technology, and the relevant terms from the field of cost accounting is given. The impact of product development on costs is described, with a special focus on the cost paradox. The currently prevailing methodological approaches for cost estimation in the product development process are set out. They are classified in the scientific context with a schema being introduced to reduce the vast number of methods to a single basic structure. Furthermore, an overview of recent work in the field of process planning is given. The focus is on traditional process planning, in particular, and on how process planning is supported by computers. The section concludes with the shortcomings and potentials of concepts and tools for cost estimation and process planning with regard to the objectives of this thesis and outlines the requirements for a cost estimation model. The first part of the thesis ends with an introduction to knowledge processing in general and the methodology of case-based reasoning (CBR) in particular.

The second part presents the model for cost estimation in detail design. Here, the rough concept for cost estimation with a focus on detail design is discussed. The aim to estimate the production costs is decomposed into the estimation of direct manufacturing costs, the provision of prices to be paid for outside company activities and the computation of overhead costs. The estimation of direct manufacturing costs based on product, process, and resource integration is subsequently further detailed. For this reason, partial models – product, process, and resource – and their integration are described. The estimation of direct manufacturing costs necessitates a vast amount of manufacturing knowledge, which is scarcely explicit and chiefly tacit. Hence, the acquisition, representation, and processing of manufacturing knowledge is the challenge this thesis deals with. Therefore, the representation of generalized manufacturing knowledge and the (re)use of specific manufacturing knowledge are further detailed. Especially the retrieve and reuse steps of the CBR methodology are described with regard to cost estimation in detail design: knowledge about manufacturing similarity is required and used to retrieve the most similar case in the case base. The section describes how to manually define and to automatically determine manufacturing similarity. The second part of the thesis brings together the individual components of the cost estimation model for direct manufacturing costs.

The third part sets out the implementation and application of CABACO, finally drawing concluding remarks on the research work presented in this thesis. A use case diagram is developed and the use cases of this diagram are further detailed to describe the prototypical implementation. The application of CABACO is demonstrated by means of an example. Furthermore, the reengineering of the processes in product creation, which are a consequence of a system for design-concurrent cost estimation are depicted. The third part of the thesis ends with a look at future work.

The cost estimation model enables the integration of cost-based decision-making in the product development process during the design process and results in a fully automatic derivation of a detailed and differentiated cost structure. The estimation of direct manufacturing costs is based on the evaluation of alternative sequences of manufacturing operations, their resource consumption, and financial consequences. The manufacturing knowledge required for this (generalized or specific domain knowledge and knowledge about manufacturing similarity) is separated from the processing knowledge, i.e. manufacturing knowledge can be added, deleted or modified at run-time. For the processing of specific knowledge, the methodology of CBR is adapted to the needs of cost estimation: In the prototypical implementation not only is it utilized as a retrieval and reuse strategy, but the definition and assessment of manufacturing similarity are also elaborated. The ease of acquisition and maintenance of manufacturing knowledge emphasize the practical value of this thesis. The model for cost estimation in detail design contributes to the design of production cost minimum products and accelerates the product development process with a decrease in product development costs.

Table of Contents

PART I: Framework

Chapter 1

Motivation

1.1 Cost Estimation

With the innovation potential of a company considered to be a measure of its success in the market, the development of innovative products has become one of the core activities in industry. Designing desirable products with a high level of functionality and quality in an ever shorter time – simultaneously ensuring the product's profitability – is a demanding goal. Thus, fulfillment of functionality requirements and assuring high quality are prerequisites for successful products; yet they alone are not sufficient. The selling price and an early market launch are factors which now serve more and more to differentiate products. The latter necessitates a reduced time-to-market and accelerated innovation cycles in order to faster respond to changing market demands and adapt to competitive situations. Therefore, the product creation processes deployed within modern manufacturing organizations are feeling the grave impacts of the demand for drastic reductions in time-to-market and cuts in development and production costs. On the other hand, developing state-of-the-art products such as today's automobiles is growing increasingly complex.

Apart from a few isolated cases (e.g. luxury goods), the selling price of a product is predefined by the market because customers are not willing to pay more than a certain amount of money for the product. Consequently, the achievable profit is dependent on the product costs. Logically if the costs deduced from the predefined selling price are not met, product profitability cannot be ensured. Thus, industry has recognized that it is insufficient to do accounting in retrospect. Rather there is a need to proactively and continuously analyze and control product costs. Cost management – the precise and systematic control of costs – has become paramount (Franz and Kajüter, 1997). Product cost management is the framework for the analysis and control activities to proactively influence and reduce product costs.

Opportunities to exert an impact on costs are greater in product development than in job planning or production. To maintain profitability and increase competitiveness, enterprises today are faced with the necessity to cut product costs, in particular by developing products which will incur lower production costs. Lindemann (1998) states that the who is able to compare design alternatives has the possibility to launch better products. Design decisions favoring the alternative with the lowest costs are typically taken on the basis of the designer's experience: and designers mainly use rules of thumb and educated guesses. Sometimes, if a more detailed cost estimate is needed, domain experts of downstream processes are commissioned with this task. However they typically use their experience in an expensive and time-consuming elaboration of the future production process and cost rates of the resources applied in an attempt to derive a monetary value for an anticipated process plan.

It is self-evident that decisions made during product development gravely impact product quality and productivity. Traditionally, products have been developed on the basis of experience and trail-and-error methods. However, the way designers make cost estimations has to be controlled otherwise this will lead to costly mistakes.

1.2 Key Objective

Presently, the design engineer is insufficiently supported in the development of cost-optimized products due to the lack of an appropriate tool for design-concurrent cost estimation. Taking into account the great responsibility the designer has to cut costs (Ehrlenspiel et al., 1999), this is an unsatisfactory situation. Design-concurrent cost estimation offers great money-saving potential and makes it possible to increase profit in the future. For this reason there is a need for an adequate methodology and IT tool which can support the design engineer in accelerating development of cost-optimized products and at the same time fulfill all the technical requirements set.

This thesis reports on the concurrent estimation of production costs of discrete mechanical parts in detail design.
Three sub-objectives are derived from this topic:

- Estimation of direct manufacturing costs.
- Provision of prices for rough parts and external services.
- Computation of overheads.

The focus is placed primarily on the estimation of direct manufacturing costs with the scientific value contained in the model for this sub-objective.

The purpose of this research is to overcome the drawback of insufficient support of the design engineer and to elaborate and prototypically implement a concept for design-concurrent cost estimation allowing early cost estimates to be obtained. Furthermore, not only should it be possible to visualize cost structures, but it should also allow design alternatives to be benchmarked with respect to costs. In this way, design decisions can be taken based on well-founded cost information. With this approach, knowledge related to production costs becomes available to a broader user group and to the design engineer in particular. Finally, the model for cost estimation is a building block for product cost management and design-for-X.

Two major benefits are expected:

- Contribution to the design of production cost-minimized products.
- Acceleration of the product development process at hand, causing a decrease in product development costs.

Because a concept is only applicable inside the model space it is designed for, the following paragraphs will further define what lies within and outside the scope of this thesis.

There are two main premises:

- The manufacturing methods and the resources are known.
- The CAD product models of the finished part and, if necessary, of the rough part are assumed to be given. They are geometrically and technologically fully specified and are adequate for a certain feature-based modeling method.

This research will not deal with situations where there are uncertain or incomplete product data: instead, the CAD product model is assumed to be complete. It reflects an abridged but true model of the future real product. If design alternatives are to be compared, the design engineer has to generate – or at least – specify CAD product models. This differs from the idea of the what-if system whose function it is to modify key parameter values, e.g. the material, with the model subsequently newly built up automatically (Lutters, 2001). This also differs from the notion presented in Leibl (1998) and Leibl et al. (1999) to automatically generate the most cost-favorable design variant.

The research in this thesis is described in terms associated with discrete mechanical products. The examples applied to explain the concept are largely related to metal-cutting and machining methods such as drilling and milling, which are used to depict resource consumption. However, this is not done because the methodology is confined to this kind of manufacturing method, but instead because this area offers a sound and surveyable overview.

The timing of production is neglected, i.e. multiple use of resources at the same time is not excluded and outsourcing or overtime allowances resulting from the exceeding of capacity limits are not taken into account. The hourly rates assigned to the manufacturing equipment and the cost rates of the operational and auxiliary supply are presupposed as being given as well as the overhead rates. Thus, the impact of a certain design alternative on cost rates, which is of interest in simulation-based cost analysis – as, for example, described in Baier (2002) and VDI 3633-7-2001 – and finally on the cost estimate itself is not considered.

1.3 Outline of the Thesis

This thesis is divided into three parts: the framework, the cost estimation concept, and the application of CABACO. In the first part, the product creation environment and knowledge processing are analyzed. The second part presents the model for cost estimation in detail design and the third part sets out the implementation and application of CABACO, finally drawing concluding remarks on the research work presented in this thesis.

Chapter 2 provides an overview of the product creation environment including feature technology, costing, product cost management, cost estimation, and process planning. The section concludes with the shortcomings and potentials of concepts and tools for cost estimation and process planning with regard to the objectives of this thesis. In chapter 3 knowledge processing in general and the methodology of case-based reasoning (CBR) in particular are introduced since they are closely related to the hypothesis of this research work. Chapter 4 develops the rough concept for cost estimation with a focus on detail design. Here, the aim to estimate the production costs is decomposed into the estimation of direct manufacturing costs, the provision of prices to be paid for outside company activities, and the computation of overhead costs. The intention of chapter 5 is to elaborate the cost estimation model for the direct manufacturing costs. The prototypical implementation of CABACO is set out in chapter 6 with an illustration in the metal-cutting domain presented in chapter 7. Chapter 8 concludes the research work presented and ends with a look ahead. The figure below sets out the outline of the thesis.

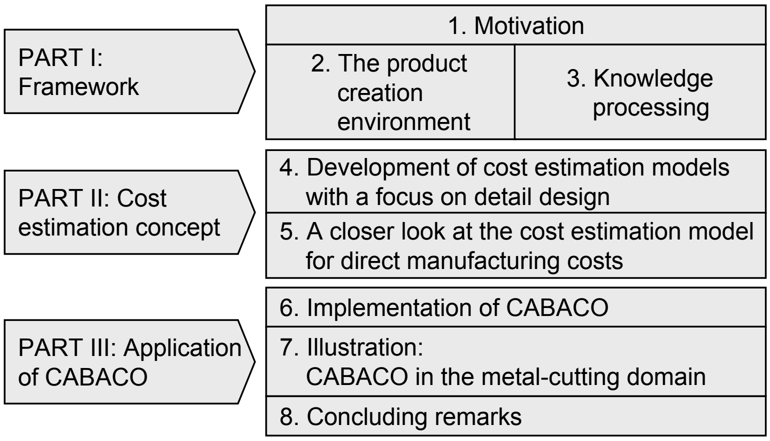

Figure 1.1: Outline.

Chapter 2

The Product Creation Environment

2.1 The Product Creation Process

Industrial enterprises differentiate three core processes in their operations: product creation, order processing, and enterprise management. The product creation process (PCP) covers the entirety of operations from the first conceptual idea to the completion of the product; order processing is dealing with the area from the very first product idea to delivery and after-sales services. Enterprise management is concerned with strategic and operational orientation, enterprise organization, and workflow optimization.

In the following, a largely industry-independent and generic overview of the mechanical engineering PCP is provided. The focus is placed on this initial phase of the product life cycle, neglecting the subsequent usage and disposal. Product creation arises in response to a perceived market need. If it is to lead to technically and economically viable products, there has to be a promising product idea (Pahl and Beitz, 1996). Product planning systematically searches and singles out seminal product ideas (VDI 2220–1980), gathering information from both the market and the company itself. It provides a product brief that describes the desired functionality and specific characteristics of the new product. The subsequent stages of the PCP – product development, job planning, and production – pursue and mature this idea.

Usually, there is no straightforward sequence in the PCP: Feedback to previous stages necessitates iterations so that carrying out activities in a predefined sequence is a substantial disadvantage. Concurrent engineering aims at developing the product and its accompanying processes simultaneously (Sohlenius, 1992). The idea of concurrent engineering is two-fold:

- To put activities in parallel.
- To integrate and overcome "over-the-wall" procedures by replacing the traditional functional organization with project-based organizations.

The benefits are improved product quality and accelerated innovation cycles. Now thanks to concurrent engineering, there is no longer a clear segregation between the activities in PCP; instead they are intertwined. Since terminology in PCP is manifold, the figure below denotes the most important stages and activities for this thesis.

Product development synthesizes geometric, technological, and economic demands in the product model, which is the medium used for communication in product creation. A product development process typically consists of three activities (Suh, 1990):

- Establishing product development objectives to satisfy customer demands.
- Generating ideas to create solutions.
- Analyzing the solution alternatives and selecting the one that provides the best fit to product development objectives.

The product model in detail design reflects an abridgement of the actual (physical) product and comprises the information needed to create the product, i.e. geometric and technological (tolerance, surface roughness, etc.) information.

This thesis applies the term model as used by Stachowiak (1973): a model is an image (Abbild) of something and an example (Vorbild) for something. Thus, a model is a simplified representation of the real world and considers solely the properties that are of importance for the problem under consideration. That means that properties which are not related to the current problem are neglected.

From the birth of the product idea to the completion of the design activities, the product becomes more and more precisely defined and is finally fully specified. Thus, product development is made up of different phases, each pursuing a distinct objective. During early phases of product development, decisions concerning product characteristics are made. The specifications generated are elaborated to ensure that technical requirements are met. Both the product geometry and supplemental product information are gradually refined and fleshed out. The product description is handed over to downstream job planning.

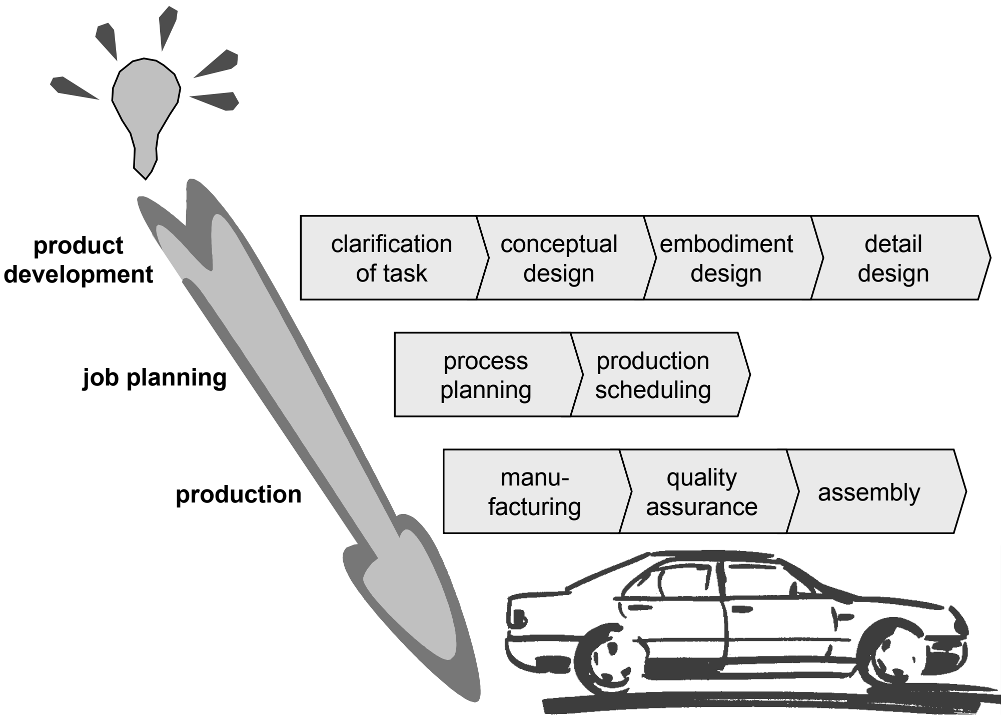

Figure 2.1: Model of the Product Creation Process.

Various activities within product development, e.g. part design and packaging analysis, interact with each other. Requirements that result from packaging have to be woven into the product model. The objective of design-for-X is to meet the various, sometimes contradicting, requirements. The *X* stands for manufacturability, assemblability, disassemblability, recyclability, reliability, aesthetics, performance, ergonomics, costs etc. The activities of product development are described in more detail in VDI 2222-1–1997 and Pahl and Beitz (1996). According to the authors, there are four phases of product development:

- Clarification of task (planning).
- Conceptual design.
- Embodiment design.
- Detail design.

Conceptual design aims at elaborating a basic engineering structure. It sets up functional structures and searches for suitable physical principles. Embodiment design maps this basic structure onto a realizable structure consisting of a number of sub-modules and components. Finally, detail design further refines each sub-module, component, and part until a sufficient level of fine-grainedness is achieved. Engineering analysis investigates the results of the design against functional specifications, e.g. strength, vibration, and noise.

As a rule, the different stages are not strictly sequential but involve various iterations. The reasons for this lie in the contradicting objectives the product has to fulfill. There are inherent relations between the parts of an assembly and between the parts or the product and the resources applied to manufacture and assemble them. In the automotive industry, for example, several iteration loops are gone through including the building of prototypes (VDI 2221–1983), simultaneously pursuing a continuous optimization of the product and the processes.

Depending on the type of engineering design, activities in the PCP may vary. Pahl and Beitz (1996) distinguish between three types of engineering design:

- *Original design* creates new basic principles for a system.
- *Adaptive design* adapts a known system to a new or a changed purpose.
- *Variant design* changes size and/or arrangement of parts or assemblies within predefined limits, not modifying the principle solution.

The proportion of each type of engineering in a company is not a technical but an economical issue and depends on customers' demands and on the company's particular strategy.

Original design runs through the four standard phases of product development whereas adaptive design skips the step of retrieving physical effects and variant design encompasses only embodiment and detail design. The development of a new passenger car, for example, belongs to the overall area of variant design yet, for components and subcomponents, applies adaptive and original design.

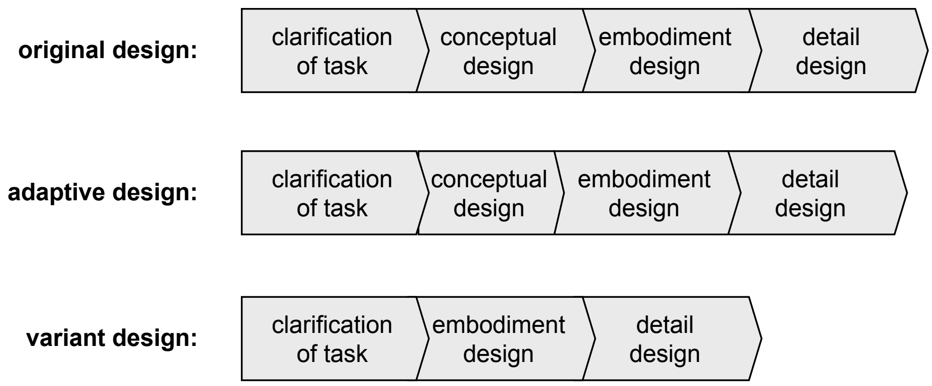

Figure 2.2: Types of Engineering Design.

Job planning is positioned between product development and production (Dubbel and Beitz, 1995). It is the aggregation of all the activities which apply the product model, elaborate the processes and the resources, and schedule the point of production. Dubbel and Beitz (1995) distinguish between two fields of activity: production planning and production control. Both process planning and production scheduling belong to production planning. As a technology-oriented synthesis task, the former comprises all non-recurring activities. It is largely independent of an exact point of production. The latter encompasses all those activities necessary to execute an order. Production scheduling concerns the short term allocation and sequencing of jobs to and on resources on the shop floor so as to optimize overall performance objectives (Giebels, 2000).

Process planning, which is the key activity in production planning, includes manufacturing, assembly, and inspection planning tasks. In the manufacturing realm, process planning – or more precisely manufacturing planning – defines the process that transforms the unfinished part into the desired finished part (Halevi and Weill, 1995). To do so, processes and their sequence of operations, resources, e.g. machine tools and fixtures, are determined. Also, technological parameters such as feeds and speeds for the cutting processes and the relevant parameters for forming processes are specified. In-process models with discrete geometric dimensions and tolerances are defined. The process planner selects these candidate processes and resources, respecting the constraints imposed by the design engineer. Yet process planning elaborates a technologically optimal solution without considering the available capacity (Halevi and Weill, 1995).

Production scheduling is triggered by acceptance of an order and is necessary to execute an order. It specifies the exact point of production and the machine tools, cutting tools, testing aids, etc. to be employed. The type of machine tool in the process plan is replaced by a concrete instance of this machine class.

Finally, production is to manufacture and assemble the product with respect to optimal technological and economic conditions. This includes quality assurance. DIN 8580–1985 identifies six major categories of manufacturing methods: creative forming, reforming, separating, joining, coating, and changing the properties of materials. This thesis applies the term manufacturing method as defined in this standard but differentiates (in conformity with the automotive industry) between manufacturing and assembly, which together shape production. Different types of manufacturing methods and manufacturing systems, e.g. flexible or cellular manufacturing systems, are applied to economically produce the product depending on its complexity and the desired lot size (small batch, large batch, and volume production).

This section has given a rough, generic description of the product creation process. In its level of detail the PCP will vary considerably depending on the particular type and degree of complexity of the product.

2.2 Feature Technology

This section introduces feature technology and feature linking. It does not intend to go into detail by giving a generic definition of what a feature is, but will instead provide the common understanding necessary for this work. Shah and Mäntylä (1995) offer a deeper insight on feature technology while VDI 2218–1999 summarizes the state of the art in this field. Salomons (1995) describes features as elements that link design and manufacturing and Geelink (1996) depicts features (e.g. holes, slots, pockets) as elements with some engineering meaning. Haasis et al. (2003) go beyond this and target the integration of product, process, and resource, applying feature technology as the underlying backbone. Katzenbach et al. (2001) elaborate on features in automotive engineering.

The following depicts feature-based detail design from the user's point of view. Advanced feature-based CAD applications provide predefined feature classes that can easily be instantiated in the CAD product model. When instantiating a design feature (DF) – a blind hole, for example – the design engineer establishes values not only for geometric parameters, e.g. diameter and depth, but also for technological parameters such as the dimensional tolerance of the diameter.

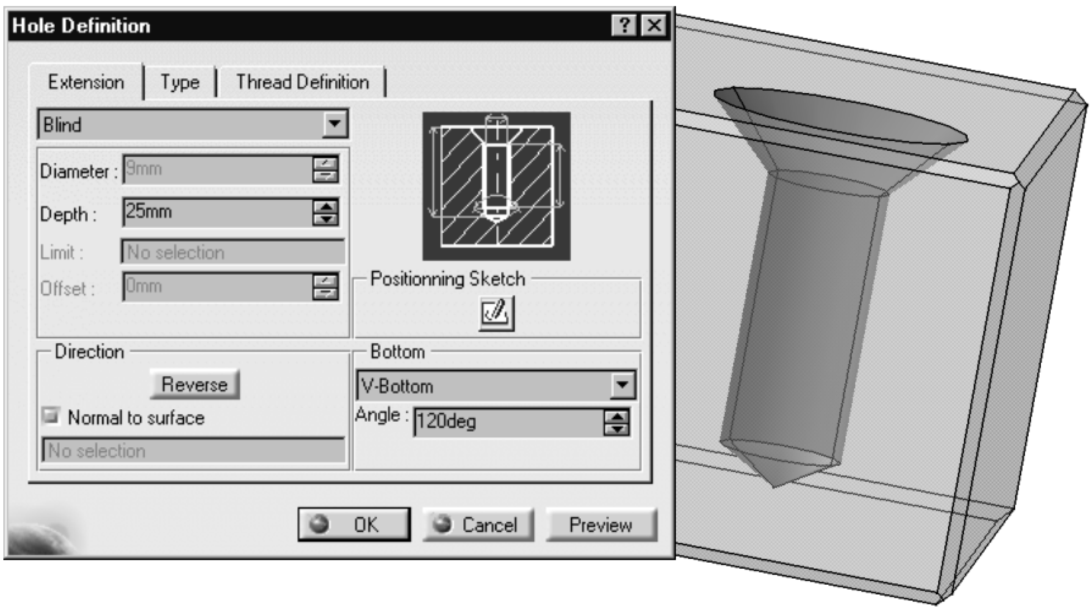

Figure 2.3: Design Feature Instantiation in CATIA V5.

In addition to the geometric and technological descriptions, a large variety of supplementary information can be linked to a feature (Haasis et al., 2003): for example, experience on using the feature (Haasis et al., 2001a). After the position and orientation have been selected, the feature is instantiated and displayed in the CAD product model. Then, if the model has to be reworked, the parameter values can be modified easily. This procedure of instantiating and modifying significantly speeds up product development. Thus, a DF can be seen as a solution pattern to optimize recurring tasks in product development. It overcomes the tedious, purely geometric construction of product models and the lack of design intent. Each DF class may be organized according to its function, enabling intuitive product modeling. In particular, the functional view as perceived by the designer is supported, thus promoting the expert's creativity (Haasis et al., 2003; Katzenbach et al., 2001).

Furthermore, the feature classes are not restricted to being predefined, but may be extended by user-defined feature classes that can be instantiated similarly. This allows commercial CAD applications to be customized. In the automotive industry, for example, DF classes for spark plug and glow plug mounting holes (Haasis et al., 1999) and for car body features (Mbang et al., 2003; Mbang et al., 2002a,b) are conceivable.

Yet, it is not only detail design that can benefit from feature technology, but also the wide variety of applications in the PCP. This includes assembly design, process planning, and engineering analysis methods such as stress evaluation using finite elements, tolerance analysis, collision detection, and manufacturability evaluation, to mention only a few. However, as each step requires specific information to fulfill the diversity of the tasks belonging to it, each process step application has its own perspective of the product model. In process planning, manufacturing features (MF) (Haasis et al., 2003) and inspection features (IF) (Haasis et al., 2000a,b; Ciesla, 1997) are applied. The usage of features in the various process step applications is similar to that described above in design: predefined feature classes, e.g. drilling or milling MFs, are instantiated in the model.

Since MFs invariably refer to manufacturing processes, the identification and formalization of MFs is intrinsically bound to the identification and formalization of the underlying manufacturing processes (Shah and Mäntylä, 1995). Neither in design and process planning nor in the other process step applications is there a universal set of features. Because of the different perspectives in detail design and process planning, the parameters of DFs , MFs, and IFs e.g. through hole DFs and drilling MFs, are not necessarily the same: attributes are application specific. To give an example: while the breakthrough is of interest when drilling through holes, it is not of interest in design, therefore being no attribute of the through hole DF. To sum up, DFs are elements related to a part's function, whereas MFs are elements that refer to manufacturing processes.

The main benefits of feature technology mentioned in literature are manifold:

- Features are objects with semantics (Haasis et al., 2003).
- Feature libraries accelerate the design process (Katzenbach et al., 2001; Shah and Mäntylä, 1995) and enable enhanced product quality (Katzenbach et al., 2001).
- Parametricity and associativity (inherent to features and feature models) propagate design changes (Shah and Mäntylä, 1995).
- Feature classes based on the function to be mapped ease design and lead to intuitive product modeling (Katzenbach et al., 2001).
- Features integrate computer-supported process chains (VDI 2218–1999).

The most crucial drawback of domain-specific feature types such as DFs, MFs, and IFs is the logical separation of the various feature-based models (Zimmermann et al., 2002a; Geelink, 1996). Shah and Mäntylä (1995) mention that an automatic derivation of one feature-based model from another is very desirable. Feature linking – feature mapping together with the generation and maintenance of persistent links between the mapped instances – tackles this issue and bridges the gaps between the multitude of different feature types (Zimmermann et al., 2002b). For example, a DF is mapped into an MF, and the link between both instances ensures associativity in both directions.

ULEO (universal linking of engineering objects) (Zimmermann et al., 2002a,b) seems to be the most promising approach for feature linking. It consists of a meta-taxonomy of relation types (MTRT) and a unified model of engineering objects (UMEO). To expand the approach beyond features to any object relevant in engineering, e.g. assemblies, parts, and surfaces, Zimmermann et al. (2002a) introduce the more abstract term engineering object (EO). Thus the next two paragraphs present ULEO as described in Zimmermann et al. (2002a,b) in more detail.

UMEO applies an object-oriented feature modeling: features are arranged in feature classes and can be instantiated into integrated partial models. To reduce redundancy according to the object-oriented paradigm, attributes which are common to several feature classes are extracted and conglomerated to a new feature class, which is termed abstraction. This parent class is related to its child feature classes with a *kind-of* relation. Feature classes on the bottom level (leaf nodes) can be instantiated whereas abstract classes located higher in the hierarchy cannot. UMEO is built up in such a way as to easily extend the model.

The MTRT contains all types of relationships between the EO classes and/or instances. Examples for relation types are inheritance, aggregation, and engineering object relations (EOR). For the latter a distinction is made between informational EORs (IEOR) and generative EORs (GEOR). IEORs describe logical relationships between EO classes, e.g. an *is_machined_as* relation between *m* DFs and *n* MFs. GEORs comprise knowledge for automatically instantiating EOs and EORs. The figure below illustrates the concept of ULEO.

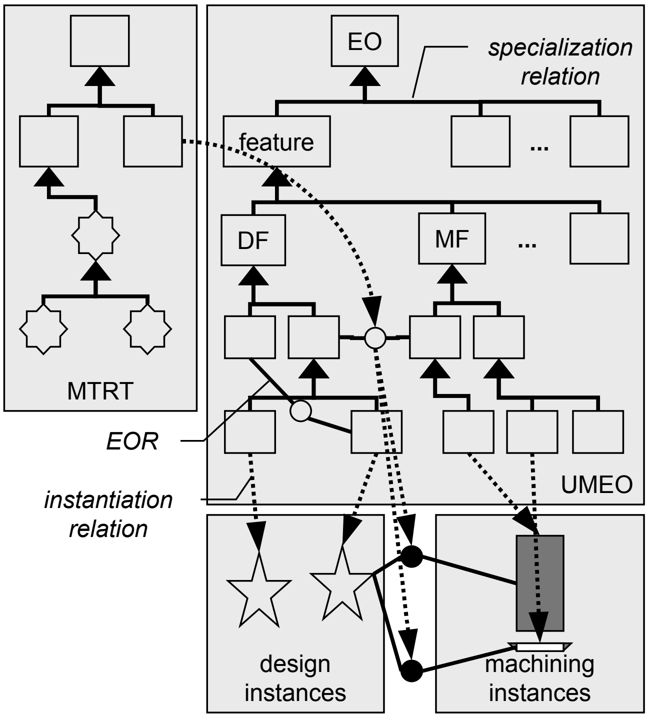

Figure 2.4: Concept of ULEO.

UMEO is an example of an integrated PPR (product, process, and resource) model that consists of various interrelated partial models and includes all the information about the product, the processes, and the resources which are necessary in the context of the PCP. Instance models, i.e. ProSAp (process step application) models, derived from UMEO are examples for integrated, distributed PPR instance models.

Figure 2.5 depicts how features in different process step applications – product development and process planning – are instantiated and how a persistent link between the instances is maintained assuming that the partial models are integrated.

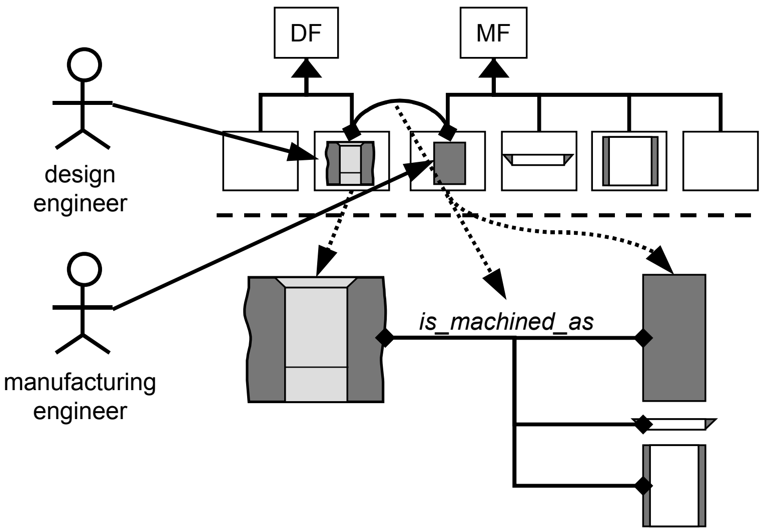

Figure 2.5: Instantiating a Through Hole DF and Its MFs.

2.3 Cost Factors

2.3.1 Terminology

For a better understanding of the content of this work, a short overview of relevant terms from the field of cost accounting is set out below.

Cost accounting is concerned with the calculation of product costs (Drury, 2000). Product costs include the costs necessary to effect production of the product (Blocher et al., 2002). Cost accounting serves to control a company's profitability and is made up of three key tasks (Olfert, 1999):

- Ascertaining cost types by gathering costs and classifying them into certain categories.
- Tracing the costs to the cost centers where they are incurred.
- Assigning the costs to cost objects.

The term cost means the monetary value of resources used (Thompson, 1999). More precisely, it is the amount of money expended in terms of labor, materials, use of equipment, etc. to achieve a specific objective, i.e. to manufacture a product. In contrast, the price is the monetary value charged for a product or a service (McLeod and Hanks, 1985).

Costs may be classified according to various criteria and from different perspectives. In cost accounting, costs can be divided into two categories: direct and indirect costs (Drury, 2000). The term overhead is synonymously used for the latter. Direct costs can be accurately traced to a particular cost object, i.e. an item such as a part for which a separate measurement of costs is desired, for instance, whereas indirect costs cannot (Blocher et al., 2002). They are usually accrued collectively through several cost objects and are therefore assigned to cost objects using cost allocations (Blocher et al., 2002; Drury, 2000; Olfert, 1999). Cost allocation is the process of assigning costs when a direct measure does not exist. Thus, in the first stage indirect costs are directly traced to cost centers and the total costs accumulated in each cost center are then assigned to cost objects using a separate allocation base for each cost center. Cost centers (or cost pools) are meaningful groups into which costs are collected. A cost driver is any factor that has the effect of changing the level of total costs for a cost object (Thompson, 1999).

While a variety of cost accounting methods are employed, they differ primarily with regard to time and scope. In point of the extent of the costs assigned, full costing captures the individual cost elements – direct and indirect costs – and assigns them to cost objects. Direct costing, in contrast, assigns only the costs directly incurred to cost objects. With respect to the different points in time (past, present or future), cost accounting differentiates between costing based on actual, normal or budgeted costs. The costing applied may, of course, be a combination of various aspects.

2.3.2 Costing

Depending on the main tasks as set out above – ascertaining costs, tracing costs to organizational structures, or assigning costs to products or services – cost accounting is split up into cost type accounting, cost center accounting and cost object accounting (Olfert, 1999). The procedure of cost accounting is independent of time (actual, normal or budgeted costs) and scope (full costs, partial costs) (VDI 2234–1990). Typically, once a year, the Controlling department determines the cost rates for the resources and the overhead rates based on information from cost accounting.

Cost type accounting (CTA) is the starting point in cost accounting, building the foundation for the subsequent activities of cost center accounting (CCA) and cost object accounting (COA). CTA answers the question *"Which costs have accrued?"*. It registers all the costs incurred within a period and classifies them according to their type, e.g. direct material, direct labor.

Cost center accounting answers the question of *where* the costs have occurred within the value chain. It takes over the indirect costs from CTA which cannot be directly assigned to cost objects because they are common to several cost objects. Costing rates are applied to break down the costs to the cost centers.

Cost object accounting records *for which object* costs have occurred and takes over the direct costs from the CTA and the indirect costs from the CCA. A subgroup of COA is cost object unit accounting (Olfert, 1999).

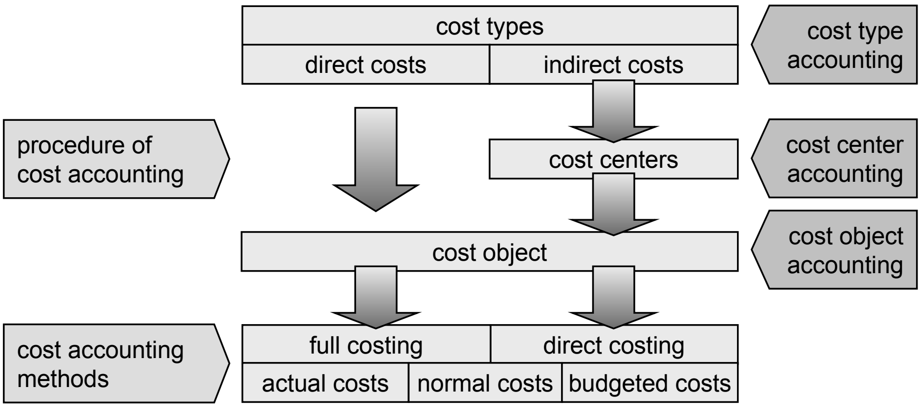

Figure 2.6: Cost Accounting (VDI 2234–1990).

Since companies differ in their product structures there are several modifications of cost object unit accounting, e.g. job order costing (non-differentiated and differentiated), costing based on the calculation unit of a machine's hourly rates. They are both appropriate for job and volume production of heterogeneous products. Differentiated job order costing splits the indirect costs up into several types, e.g. material overheads, manufacturing overheads. Job order costing accumulates costs for individual jobs or lots (Thompson, 1999). Bases for further calculation are, for example, the direct material costs. Figure 2.7 depicts a typical differentiated job order costing schema. The objective is to determine the total product costs and the selling price.

If necessary, the costs in this schema can be further broken down into recurring costs, e.g. for raw materials, and non-recurring costs, e.g. for raw material qualification.

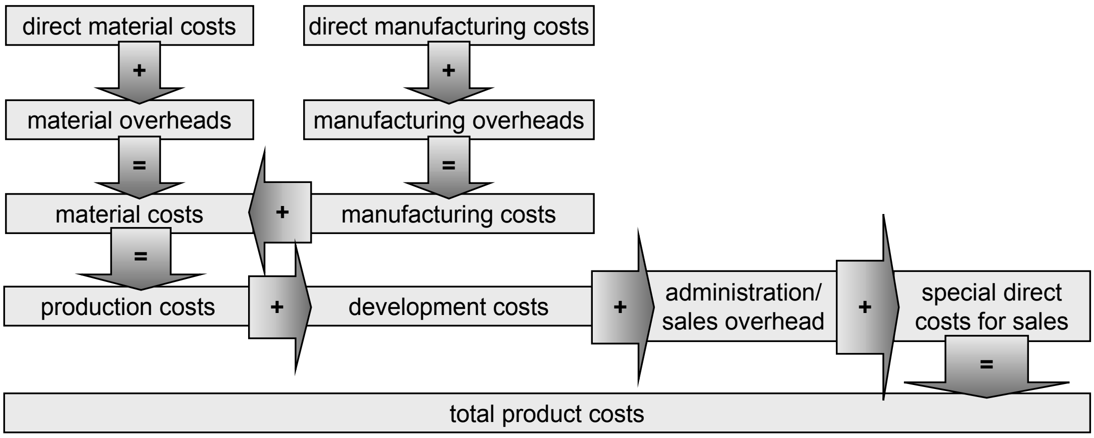

Figure 2.7: Differentiated Job Order Costing.

If the degree of automation in production is high, job order costing is not able to guarantee the sufficient assignment of costs (Olfert, 1999). In job order costing, the overhead costing rate for manufacturing determined in cost center accounting is applied for each cost object. However, this results in undifferentiated burdening of the cost objects. Costing based on calculation units breaks down manufacturing costs even further, assigning them more on the basis of use than differentiated job order costing does. Production costs and total product costs are determined analogously to differentiated job order costing (Olfert, 1999).

Depending on when the unit-of-output costing is established, specialists distinguish between three different types of calculation in cost accounting (VDI 2234–1990):

- Pre-calculation.
- Intermediate calculation.
- Post-calculation.

Pre-calculation targets the determination of future costs before the actual production has started and thus allows cost-based decision-making. The costs may already be fixed or, as is the case when calculating a bid, only roughly outlined. In pre-calculation, business-oriented approaches and purely engineering methods are distinguished. Yet, frequently, a clear separation is not possible (generative-analytical approach) as one approach builds on the other and vice versa. Statistical approaches and analogy approaches (engineering methods) will be addressed later on in this work. Pre-calculation applying cost object unit accounting is based on a yet unknown quantity schedule (resource consumption, i.e. machining times, resources, etc.). This necessitates anticipating the future process of production, which is a task for the engineer rather than for the businessman and is the main concern of this thesis. Pre-calculation is usually only able to access product describing data, and unfortunately such data may be incomplete or uncertain (Neff et al., 2000). In the remainder of this thesis the term cost estimation denotes engineering pre-calculation.

In contrast, post-calculation – as a pure method of cost accounting – determines the actual costs accruing, with these costs then serving as the base data for future pre-calculations. During the product development cycle, intermediate calculations are carried out and applied for cost controlling purposes. Intermediate calculations are a mixture of parts of pre- and post-calculation.

As depicted, cost accounting aggregates costs at different levels. The total product costs comprise all the costs occurring in manufacturing and assembly: in addition to production costs, this includes development costs, sales overheads, etc. The achievable profit is equivalent to the difference between the selling price and the total product costs. Life-cycle costs include all the costs accruing due to purchase, use, and disposal.

Newer concepts of cost accounting such as activity-based costing methods were developed as a solution to the problems of cost aggregation. Activity-based costing differs from traditional costing in that it applies more appropriate bases for cost allocation (Eaglesham, 1998). However, it is still retrospective rather than prospective. What the different methods have in common is that all costs are covered and none are counted twice.

2.3.3 Product Cost Management

Recognizing that accounting in retrospect has drawbacks in today's dynamic market, industry has changed its focus to proactive and continuous analysis and control of product costs. Cost management means systematically controlling costs (Franz and Kajüter, 1997). The objective of cost management is to influence the costs of products, processes, and resources – the three elements of cost management – by means of concrete actions in order to achieve suitable outcome for the company and to assure competitiveness (Franz and Kajüter, 1997). Product cost management as one kind of cost management is the framework for the entirety of analysis and control activities to proactively influence product costs. Yet product cost management and cost accounting are not mutually exclusive, rather they interwork.

Product cost management is a valuable approach empowering companies to effectively manage the product costs: it comprises activities to gather market information and match this data to product costs, to predict costs trends and to recognize when cost targets are endangered. In conclusion, product cost management enables organizations to focus on product costs as one of the crucial factors in product creation. It facilitates a deeper understanding of customer affordability and the competitors' pricing policies, a detailed quantification of costs and access to reliable cost data. Ehrlenspiel et al. (1999) provide an in-depth survey on instruments for cost management and Krause (1997) proposes combining cost management with the possibilities of virtual product development. From the perspective of this thesis the following instruments of cost management are essential to effectively perform the above tasks:

- Target costing.
- Cost estimation.
- Cost accounting.

Value analysis (VDI 2800–2000) as a further instrument for cost management will not be considered in this thesis.

It is not the supplier or manufacturer who fixes the price of a product, but the market. Target costing determines a hypothetical selling price in conformity with market demand. While it may also be applied for product cost management, this thesis regards target costing as an instrument for cost management.

According to Seidenschwarz (1991), target costing involves three elements:

- Target cost detection.
- Target cost breakdown.
- Target cost tracking.

The process of determining target costs is split up into two steps: target cost detection and target cost breakdown. Numerous methods exist for target cost detection, e.g. market into company, in and out of competitor (Ewert and Wagenhofer, 2000). The basis for target cost detection is the target price, i.e. the price the customer is willing to pay. The target profit is subtracted from the achievable target price to derive the allowable costs, i.e. the costs that must not be exceeded. Drifting costs are the costs that production would incur based on existing technologies. Usually, the drifting costs are higher than the allowable costs. Therefore, in practice the allowable costs are not directly taken as the target costs. Depending on competition and a company's strategy, the target costs aimed at are set between the allowable costs and the drifting costs. Thus target cost breakdown apportions the target costs to components and parts. This cost target is then deemed established. In the majority of cases a great deal of effort and various iterations are required to reach this target.

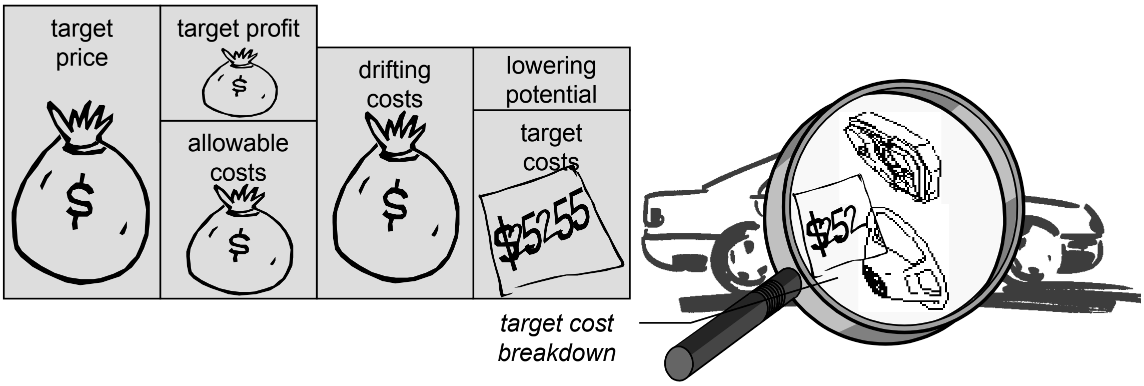

Figure 2.8: Target Cost Detection and Target Cost Breakdown.

Which actions can be taken to meet target costs and to cut product costs? Possible measures are, for example, training of personnel and implementing organizational activities. Through training, design engineers have become sensitive to cost issues and acquire more in-depth knowledge about costs and cost relations. The formation of interdisciplinary teams can optimize the availability and incorporation of cost knowledge in the PCP. These possibilities have been recognized and have been introduced within the companies. But, this general notion of cost consciousness usually leads only to qualitative statements.

However, effective control of target costs necessitates more than just qualitative statements. A quantitative statement in product development and job planning is essential. With this quantitative statement a discrepancy between the actual costs and the target costs can be made transparent (Ehrlenspiel et al., 1999). However, the output of traditional cost accounting will not result in such a statement because cost calculation requires the complete technical documentation (bill of materials, process plans, and schedules), which is only available at a much later stage in the process – much too late in the game for feasible calculation of target costs. A reliable methodology for cost estimation needs to be integrated in the product creation environment. Proactive product cost management has to be supported to enable the comparison of target costs and actual costs. In this way viable possibilities to cut product costs can be identified. For this reason, this thesis focuses on design-concurrent cost estimation.

2.3.4 Cost Paradox in Product Creation

Through the synthesis of geometric and technological information, product development implicitly prescribes the production process and determines the framework for production costs. Yet the costs, in fact, accrue due to later resource consumption. As part of an exploratory study in the area of conventional and progressive dies in the automotive industry (OEM and suppliers), the aspects of cost determination, costs incurred and cost transparency have been derived (Frenzel et al., 2001a). The result of this study proved that product development accounts for approximately 60 percent of the future product costs. Purchasing, which is responsible for rough parts and purchased parts, determines 20 percent. Job planning is the driver for the remaining 20 percent. According to this study, the lion's share of the costs, namely 70 percent, is incurred in manufacturing. Of the remaining 30 percent, 10 percent are incurred by product development and 20 percent are accrued in the purchasing of rough parts and purchased parts. In the relevant literature, usually roughly 70 percent of the costs are given as fixed in product development (Ehrlenspiel et al., 1999; Binder, 1998; VDI 2234–1990; VDI 2235–1987). The exact percentage, however, depends on the type of product and on the type of production.

The graph set out in figure 2.9 illustrates the courses of undetermined costs and available cost information for original design. The proportion of undetermined costs declines considerably at the beginning of the PCP to level off at zero in the later stages of the process. The available cost information provided without appropriate cost estimation increases slightly during product development to then soar until, due to post-calculation, the overall product costs are finally known. The divergent courses of undetermined costs and available cost information (without cost estimation) form the cost paradox in product creation. In the absence of proper cost estimation, the available cost information corresponds largely to the accumulation of the costs incurred by that point in time in the PCP. The desired state is exactly the opposite: the curve should initially rise sharply to subsequently flatten out at a very high level.

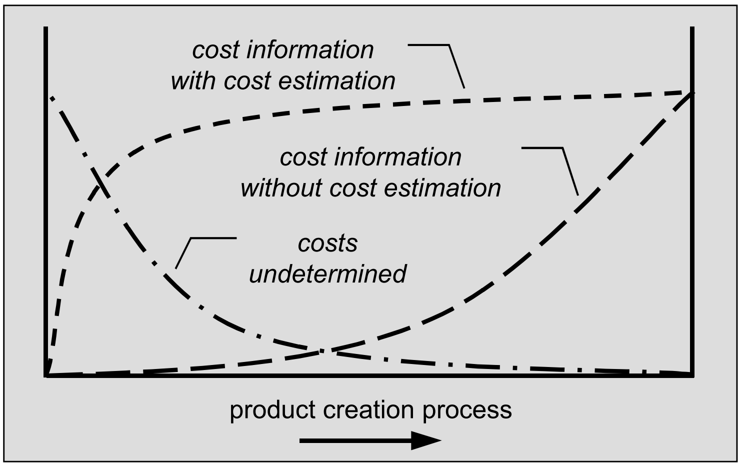

Figure 2.9: Cost Paradox in Product Creation.

Depending on the type of engineering design, sooner or later the curves of undetermined costs and cost information will be entered there. The next paragraphs will describe some key characteristics of the types of engineering design and their impact on cost estimation. Original design is concerned with creating new basic principles for a system. The desire or need to offer the same product functionality, i.e. convert fuel to energy as in an automobile but through different technologies, e.g. combustion processes in a combustion engine or electro-chemical processes in a fuel cell, is motivated by various factors, one of which is the issue of economy. Introduction of technologies that are different from those currently deployed drives engineers to consider new manufacturing processes, necessitating that more money be spent on product and manufacturing process development, on new resources, etc. Westkämper (1997) states that, under the pressure of competition, a great deal of effort is spent in production to decrease the costs of the currently running production. Thus, the costs of many technical products follow the regularity of learning curves. If the number of products manufactured increases, the product costs decrease because of the totality of improvements.

Even if the application of another technology may lead to cuts in the production costs, the overall product costs may still increase because of higher product development costs, outlay for necessary new investments, etc. Thus, the degrees of freedom accorded to the design engineer are very often restricted and a vast amount of product development work is expended for variant design activities. This is due to the fact that frequently the currently deployed technology is the most economical one.

If design is of type variant, product development omits conceptual design (see figure 2.2). The consequences are that the first portion of figure 2.9 is cut off, i.e. a considerably smaller proportion of the costs is undetermined. This is due to the fact that, for variant design, the physical effects of the technologies deployed are predetermined, so that the search for basic principles is skipped. Even if it is obvious that opportunities for cost improvement lie in early involvement, for variant design and adaptive design cost information for the detail design phase is needed in any case to reduce the costs in the sector remaining for the physical effect since only small savings for a single part lead to considerable savings amounting from the high number of pieces in volume production.

Thus, it is indispensable to base design decisions in product development on a reliable cost estimate. In spite of the fact that product development is the key factor in production costs, design engineers frequently make design decisions which impact product costs solely on the basis of their own experience. Design engineers have few tools and little quantitative information for support in cost estimation (Locascio, 2000). Sometimes similarity-based techniques or simple parametric functions are employed and rules for design-for-cost applied. If a more accurate and in-depth cost estimation is needed, greater effort is involved: this means that the job planning department generates a process plan as the foundation for cost estimation.

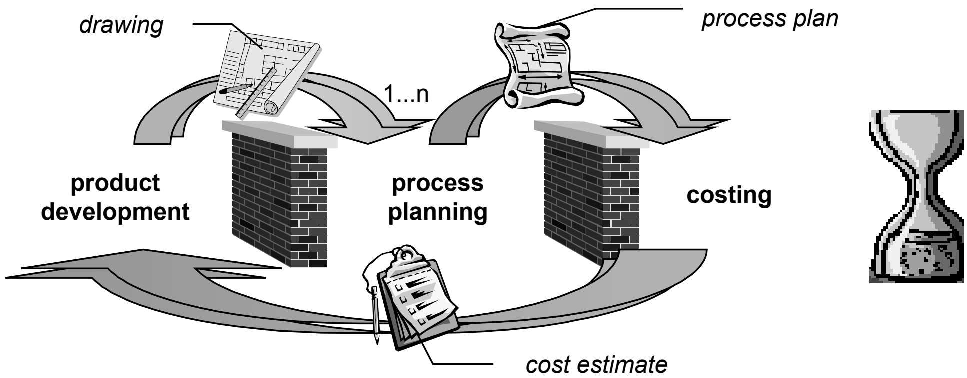

Figure 2.10: Non-Computer-supported, Analytical Cost Estimation.

But, as figure 2.10 depicts, this is extremely time and resource consuming and only few experts are able to effectively perform this process planning activity. Usually, technical drawings, process plans, and paper-based cost estimates are the basis for communication between the departments involved. These become barriers preventing the continuous flow of information and finally leading to redundant and inconsistent information. Because such an extensive feedback loop is tedious, the product model may, in the meantime, have changed essentially before vital feedback is available. As a result, the cost information has become obsolete and the effort was in vain. Efficient product cost management requires that the long feedback loop be replaced by a short feedback loop and that an integrative methodology for cost estimation be applied. Such a short feedback loop has been described in Becker (1990).

In addition to the production costs, a large proportion of the cost for use and disposal (not to be considered any further here) is also fixed since product development implicitly predetermines use and disposal (Ehrlenspiel et al., 1999; Trender, 2000). Therefore, over the complete product life cycle, product development has the greatest impact on costs.

2.3.5 Cost Estimation Approaches

Some methods for cost estimation have already been sketched. In this section, different methodological approaches for cost estimation are described and classified in the scientific context. Depending on the approach chosen, cost estimation provides either a qualitative or a quantitative result. Qualitative approaches, e.g. good/bad examples or heuristic rules (Horváth et al., 1997; Gerhard, 1994), illustrate whether any one design alternative is better or worse than another, yet they do not specify the absolute value. Thus they are not well suited if costs are to be transparently depicted or quantified, as is the case when target costs have to be pursued. For this reason, qualitative cost estimation is neglected in this work with the focus placed on quantitative approaches.

Trender (2000) and Horváth et al. (1997) distinguish between lump-sum and analytical quantitative concepts. The chief difference is that lump-sum concepts do not consider the characteristics of the production process, thus neglecting to show the cost structure in detail or differentiated in any way.

Examples for lump-sum approaches are guesstimates, i.e. guesses made by experts, accelerated cost estimation, and methods based on analogy and statistics (Trender, 2000). Accelerated cost estimation methods are simplified, easy-to-use relations between cost-effective product characteristics and costs (Ehrlenspiel et al., 1999; DIN 32992-1–1989; VDI 2225-1–1997).

Asiedu and Gu (1998) single out parametric, analogy, and detailed models for cost estimation. They describe parametric models as sets of formulae for top-down estimation that are generated by means of statistical methods to correlate costs and product characteristics. In contrast, estimation using analogy identifies a similar product and reuses the cost information. Detailed models that incorporate a bottom-up estimation depict the product creation process and analytically place a monetary value on resource consumption. Bode (2000; 1998) employs neural networks to link product characteristics to costs.

Duverlie and Castelain (1999) distinguish between intuitive, based on analogy, parametric, and analytical methods for cost estimation. An intuitive method is based on the experience of the estimator and, hence, the result of the cost estimate is always dependent on the knowledge and the experience of the estimator. The analogy method applies similarity and the parametric method characterizing parameters of the product without describing it completely. The method of scales is an example of a parametric method: it applies to products of variable size and necessitates determining the most important technical parameter, e.g. mass, to finally determine the ratio to quantify, e.g. $/kg. The analytical method evaluates the costs of a product from a decomposition of the work required into elementary tasks.

Ten Brinke (2002), Kals et al. (1999), and Liebers (1998) distinguish between variant and generative cost estimation and hybrid cost estimation, which is a combination of the two former approaches. The principle of generative cost estimation is the composition of the costs from its constituents, while variant-based cost estimation employs similar products manufactured in the past to determine the costs.

Layer et al. (2001) introduce a schema that reduces the large number of methods to a single basic structure. They distinguish between three completely different models on the basis of their theoretical foundation: statistical, analogy, and generative-analytical models. The former two are top-down approaches whereas the latter is mainly a bottom-up approach. Statistical models utilize not only sets of formulae but also neural networks to compute costs and generative-analytical models reflect a generative and analytical – in the sense of systematic – method. Figure 2.11 depicts the three models of quantitative approaches. The basic structure is elaborated in the following section.

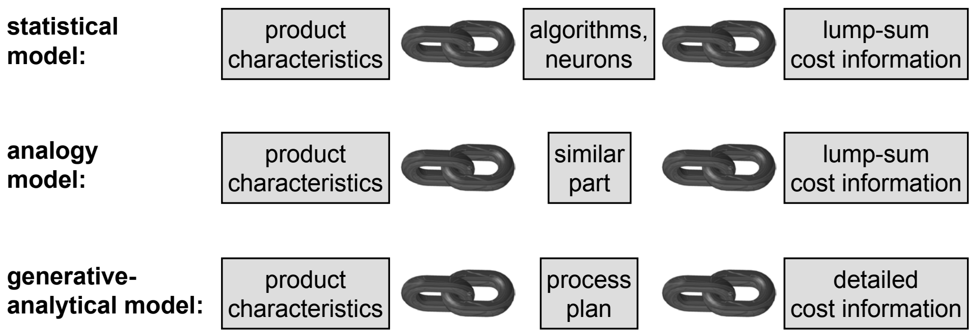

Figure 2.11: Link Between Product and Cost.

In statistical models, historic data and empirical examinations are evaluated with the objective of gaining information about the causal link between product characteristics and costs. The evaluation may be carried out using methods of regression analysis (DoD, 1999; Schreve et al., 1999) or optimization techniques (Pickel, 1988) or by employing neural networks (Bode, 2000; Becker and Prischmann, 1994a,b). Evaluation done using the two former methods leads to a parametric function with one or more variables. The complexity of the result depends on the number of cost-relevant product characteristics.

After the neural network has been modeled and trained with adequate, historic data, it can be applied for cost estimation. The input parameters of the neural network are shape-describing and semantic product characteristics; the product costs are the output parameters.

Neff (2002), Neff et al. (2000), and Asiedu et al. (2000) propose an approach to quantify associated uncertainties of cost estimations. In front load costing (FLC) both product-describing attributes and the technical and economical uncertainties are the input variables for the parametric model (Neff, 2002; Neff et al., 2000).

Cost estimation using analogy reasons from functional or geometrical similarity to a similar cost structure (Grundmann, 1994; König, 1994) with the term *similarity* describing the level of correspondence of the relevant characteristics. Information on the costs accruing during production is taken from post-calculation.

Analytical approaches depict the relevant processes of product creation in detail and derive the costs incurred, aggregating them properly. The result of the analytical approach based on a generative process plan is a detailed and differentiated cost estimation that enables specific conclusions about the cost drivers to be drawn and alternatives for adjusting product cost to be derived. Changes in boundary conditions, e.g. new manufacturing technology, new machines, etc., can more easily be considered as the model employed for calculation is newly generated. The downside is that an enormous amount of information and knowledge is needed. Thus, without computer support, this approach is extremely time consuming and difficult to carry out (Layer et al., 2002a).

Each of the three methodological approaches described uses product characteristics as input variables and returns cost information. In contrast to generative-analytical models, statistical models and analogy models determine costs in a lump-sum fashion in the vast majority of cases. But, each one of the approaches mentioned above has its advantages. Thus, depending on the relevant phase in product development and on the part of the production costs to be estimated, e.g. direct manufacturing costs and costs for the un-machined part, one of the approaches yields a best fit. For example, Frech (1998) sets out different methods whose application is contingent on the progress of product development. During conceptual design, manufacturing costs are derived using analogy, whereas during embodiment parametric functions are applied and during detail design an analytical approach is proposed. In detail design the most detailed and differentiated cost structure for manufacturing costs is obtained using a generative-analytical model (Layer et al., 2002b). Hybrid models do not purely rely on one approach rather they employ a combination of at least two of the three approaches cited.

2.3.6 Concepts and Tools for Cost Estimation

This section provides an overview of recent work in the field of cost estimation and describes typical methods deployed. The selection of examples is motivated by their applicability for and adaptability to the estimation of direct manufacturing costs in detail design. Several methods for cost estimation are introduced and elaborated in Debuschewitz (1999), Scholl (1997), Eisinger (1997), Eitrich (1996), Kümper (1996), Heine (1995), and Gröner (1991). A comprehensive survey on the state of the art in cost estimation is also set out in Layer et al. (2002a).

Endebrock (2000) and Endebrock and Welp (1999) describe cost estimation in the early phases of the product development process based on product concepts. Functional properties of the product such as forces, torques and velocity which are relevant during the use of the product are applied for cost estimation. Fuzzy rules are generated where functional and design variables are connected to cost quantities by if-then rules. The rules are set up based on human expert knowledge or data floated using automatic systems for fuzzy rule generation. Also Leidich et al. (2001) are concerned with cost estimation in conceptual design and describe a fuzzy-based method that divides a product into functions and sub-functions for which costs are represented.

Grundmann (1994) assumes that similar products have similar process plans and, thus, incur similar costs. To prepare the search for similar parts, finished injection-molded parts are specified through five characteristics and represented in the form of a point in the five-dimensional feature space. Grundmann (1994) extends the indices straightness, wall thickness, and bulkiness previously introduced by Pacyna et al. (1982) by the two characteristics logarithmic volume and number of threads. The degree of similarity is computed as a factor of the distance between the points. The bills of materials and the process plans are the input for calculation. Manufacturing times, resources, quantities, and hourly rates form the foundation on which the calculation is to run.

DEVIPLAN contains libraries with manufacturing features (form DFs in the understanding of this work), manufacturing operations, machines, and tools (Kiritsis and Xirouchakis, 2000; Kiritsis et al., 1999). The user interactively generates the geometry, adds tolerances and the surface roughness, and then selects machines, tools, and processes. The cost estimation is based on fully specified product geometry and is applicable during detail design. Because of the interactive input of data and the lack of an automated interchange of the product model from a CAD system, this approach does not permit design-concurrent use.

Leber (1995) addresses the determination of resource consumption for subsequent monetary evaluation based on the integration of product, process, and resource. Eversheim et al. (1998) describe a cost module that is founded on resource-oriented cost analysis. On the basis of the input parameters design specifications (e.g. part dimensions, materials), production parameters (e.g. parts per year, scrap rate, cycle time), and economic parameters (e.g. wages, working days per year), the process model is generated and the costs of each process step are calculated by its resource consumption. Wartzack (2001) and Wartzack and Meerkamm (2000) elaborate on the integration of a feature-based CAD product model and cost estimation. Cost estimation is based on empirically derived cost parameters to determine the manufacturing costs. The relation between product characteristics and manufacturing costs has to be represented in the knowledge base of the cost estimation tool HKB in advance. Yet, in each case, the bottleneck of acquiring the necessary knowledge is not adequately treated.

Schaal (1992), Wolfram (1994), and Steiner (1996) generate the process plan as the basis for cost estimation with rules. Schaal (1992) applies manual feature recognition. Although Wolfram (1994) and Steiner (1996) do not employ the subsequent feature recognition, they model the product in a feature-based fashion. To simplify, Schaal (1992) and Wolfram (1994) assume the following to be given:

- Any feature can be manufactured independently of all the others.
- The machining time is computed using the machined volume.

To accomplish the planning and evaluation tasks, Steiner (1996) deploys different models:

- The manufacturing model is the database that contains all the technology and resource data, e.g. machines, tools, fixtures, cutting parameters.
- The process model describes both the manufacturing processes and the linking of machines to processes.
- The cost accounting model contains the calculation method and the hourly rates.
- The planning model includes rules for the automatic generation of process plans.

The method described in Schaal (1992) is not applicable "for too complex technical fields, that means no mass production and no high-tech products as for example turbine blades". Reischl (2001) offers insight into the practical application of such a tool for design-concurrent cost estimation.

The functionality of INFOGUSS, which is part of a knowledge-based design system, is described for generative design of cast iron housings (Haasis, 1995). The manufacturing costs are split into costs for modeling, casting, and machining. Modeling and casting costs are computed using parametric cost functions that are derived through regression analysis. Then, a process plan-based approach is taken to calculate the machining costs. Yet, geometric features also contain manufacturing information. The process plan is generated using a rule-based approach. The allocation of machining operations to features is done by employing rules, which makes it more difficult to integrate experience knowledge.

Ou-Yang and Lin (1997) estimate the manufacturing costs of a design according to the shapes and precision of its features. The foundation for this approach, the feature-based CAD model is analyzed and manufacturing processes are generated using a set of rules. Manufacturing times are calculated with two underlying factors: the volume of the material removed and the specified surface roughness of each feature. Ou-Yang and Lin (1997) simplify by assuming that the surface roughness depends solely on the machine utilized. The fact that the tool, the cutting speed, and the feed rate also impact the surface roughness is not considered. A drawback to this approach is its a priori rule-based allocation.

CostDesigner from AgilTech is an expert cost estimation system for the SolidWorks CAD system (CostDesigner, 1998). The features from the solid model of SolidWorks are identified and automatically grouped in setups, for which production processes and production times are generated. Cost estimates are based on a burden rate, labor rate, load/unload time, and the setup time and are organized according to setup and feature.

Ten Brinke (2002) and Liebers (1998) deal with cost control in manufacturing. Liebers (1998) describes a generic cost control architecture. A reference model for the PCP has been developed to indicate the position of cost control in relation to other engineering tasks. The cost control architecture is decomposed into four functions:

- Cost estimation.
- Production monitoring.
- Cost calculation and evaluation.
- Cost modeling.

Several feedback loops have been distinguished, which are necessary to ensure the availability of cost-related information and cost rates. In the cost control architecture, the cost estimation function is integrated with technical planning, logistics planning, sales, procurement, etc. Cost estimation relies on company-internal or generally available information translated into cost models. A cost model is described as a cost function that relates costs to a cost carrier on the basis of the values of several cost drivers, which is a parameter that has an impact on costs. For variant cost estimation, coding is employed to establish the similarity between a part to be currently estimated and previously manufactured parts. For generative cost estimation, cost-related information is linked to basic, i.e. low-level, objects that recur in every product. Figure 2.12 depicts a redrawn version of Liebers' (1998) generic architecture for cost control.

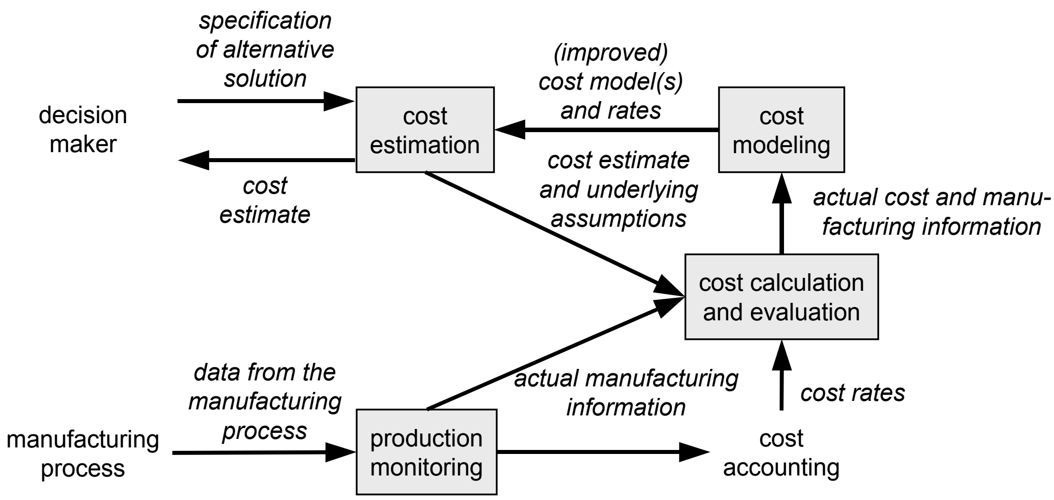

Figure 2.12: Generic Architecture for Cost Control.

The cost estimation system described in Ten Brinke (2002) is based on the generic architecture of cost control mentioned above and targets the provision of cost information throughout the whole product development cycle. Cost models are defined hinging on the cost structure, which incorporates cost types, cost functions, and cost parameters. Different cost models can be defined for different stages of the product development cycle.

Conventional IT systems are inadequate for the storage and retrieval of knowledge. Also, acquisition and maintenance are difficult in the classical expert systems. For this reason, Meyer (2001; 2000; 1998) and Stöckert et al. (1999) focus on structuring technical and economic data that is stored in IT systems during everyday work so that it may be utilized for decision support. Hence real solution patterns are stored in a database-supported case base and a product data management (PDM) system is integrated in this case-based decision support system. Generic, semantic domain knowledge supports the episodic knowledge of the case base. Products are classified using item attribute types. The concept is employed solely for decision support, not for design-concurrent calculation as the cases cannot be adapted.

The strategy for automatic cost estimation throughout the product development process as described in Rehman (2000) and Rehman and Guenov (1998a,b) incorporates the usage of case-based and rule-based reasoning. As cost estimation at conceptual design involves recalling past designs, case-based reasoning (CBR) is beneficial at this stage. The objective of the case-based design facility is to consider the incomplete description of the new design problem, retrieve a similar past design from the case base and adapt the retrieved design to satisfy the new problem description. The cost data is updated by the application of adaptation rules stored within the design models. The adaptation rules at the part level are manufacturing rules, which can be applied to construct rough process plans for the part design. At the next level, assembly rules adapt the costs. When changes are made in the CAD model, the costs are automatically updated. To sum up, case-based reasoning is employed to retrieve a similar product model completely described, and rules are employed to derive the process plan.

Frenzel et al. (2001a,b,c) introduce a generative-analytical approach for cost estimation that relies on feature technology and applies generalized and specific manufacturing knowledge to generate an in-advance process plan. An operation in this process plan is specified by the resources applied, the type of operation, and the machining time, which is subsequently monetary valued applying a cost accounting model and the hourly rates of the machines. This idea has been further developed in this thesis.

2.4 Survey of Process Planning

Process planning is of importance for cost estimation, especially when direct manufacturing costs have to be derived in detail and differentiated. Hence, this section focuses on traditional process planning and on how process planning is supported by computers. As cost estimation has to be carried out design-concurrently the focus of chapter 2.5.2 is on concepts and tools for automatic process planning.

2.4.1 Process Planning Approaches

Process planning is a wicked problem: it cannot be broken down into a sequence of sub-problems whose solutions can be combined to yield the overall solution (Shah and Mäntylä, 1995). Process planning rather is an iterative cycle in which the elements are interdependent, i.e. everything depends on everything else. In the vast majority of cases, knowledge in process planning is based on the implicit experience of the process planner (Hintz, 1996; Halevi and Weill, 1995). Paulokat and Wess (1994) maintain that human planners employ an enormous amount of heuristics and domain-specific reasoning. Halevi and Weill (1995) state that when preparing a process plan for material removal processes, a high number of solutions has to be considered. Some of the challenges in process planning they mention are as follows:

- The wide variety of processes and resources, the complex products.
- Highly sophisticated technological dependencies.
- Company-specific manufacturing technology subject to continuous improvement.

The traditional approach to process planning is founded on the process planner interpreting the product model, identifying similar parts or characteristics, and recalling past processes. In process planning, various iterations are run through with the manufacturing task manifested by the un-machined and the machined part becoming more and more detailed until single tool movements are obtained. Figure 2.13 depicts four levels of detail and the elements making up each level.

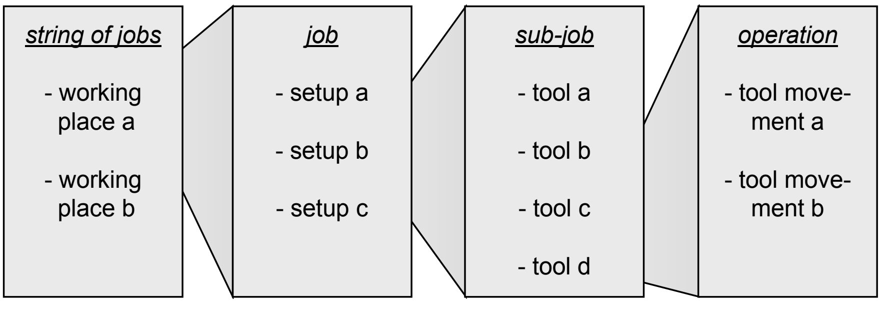

Figure 2.13: Explanation of Process Planning Terms.

A string of jobs comprises a complete manufacturing task, several workstations, and manufacturing methods. The jobs that are executed to transform the rough part into the finished part are run through in sequence. The expression *job* denotes a sequence of sub-jobs that are carried out at the same workstation. The sub-job comprises operations that are performed in a single setup; changing the setup results in a new sub-job. Again the commonality of an operation is that one tool is applied. In this work, the term *operation* is restricted to an activity that is applied to one object or to more than one object simultaneously and is usually limited by a positioning movement of the tool. In turn, an operation consists of tool movements.

Two alternatives for streamlining of process planning are conceivable: systematization, e.g. definition of standards, or automation using computer-supported process plan generation, NC-code generation, or other means. And it is the task of process planning, in particular, that is supported by computer-aided process planning (CAPP) systems and computer-aided manufacturing (CAM) systems. CAPP forms the link between CAD and CAM. CAM starts with the computer-supported generation of NC code and includes controlling machine tools. For CAPP, one distinguishes between variant, generative, and hybrid approaches. Eversheim and Michalas (2001) depict CAPP tools for each of the three approaches.

The variant approach to CAPP was first used to computerize process planning (Marri et al., 1998; Sormaz and Khoshnevis, 1997). Variant-based CAPP relies on the concept that similar parts have similar process plans. For a new part, the existing process plan of a similar part is retrieved. The retrieval is done using similarity assessment or part classification, e.g. Group Technology (GT) methods. GT classifies and codes into families of parts based upon several characteristics. A GT coding scheme maps a proposed new design into an alphanumeric code (Shah and Mäntylä, 1995). For each part family, a master process plan is defined, retrieved, and edited for a new part (Halevi and Weill, 1995). GT coding is also applied for variant cost estimation to retrieve the most similar part (Ten Brinke, 2002).

The variant approach suffers from two major drawbacks (Marefat and Britanik, 1997):

- If products vary over time, it is very difficult to find an appropriate similar part in the database that satisfies similar specifications and requires similar production.
- The variant approach takes the stability of the manufacturing environment for granted. If the retrieved process plan applies out-of-date processes, then these will also be applied for the new product.

In generative planning, process plans are created from scratch by means of technology algorithms, decision logic, or artificial intelligence (AI) techniques representing available process planning knowledge (Sormaz and Khoshnevis, 1997; Halevi and Weill, 1995). Applying AI techniques is generally referred to as knowledge-based process planning. Hintz (1996) describes knowledge acquisition and representation as the most crucial problem of generative planning. Marefat and Britanik (1997) state that generative planning hardly ever utilizes lessons learned.

Hybrid or semi-generative process planning is an intermediary approach which uses knowledge in existing plans, synthesizes this knowledge, and thus generates process plans for the new product model. The overall task of plan generation is broken up into several partial tasks. The partial solutions are aggregated to finally obtain a complete process plan.

2.4.2 Concepts and Tools for Automatic Process Planning

Various concepts for CAPP have been pursued in the past (Marri et al., 1998). Usually, CAPP systems, which offer some kind of automation, force expert process planners to formalize manufacturing knowledge in rules or in a kind of master process plan. This presupposes that manufacturing knowledge can be acquired, represented, and maintained.

Subsequently, concepts and tools for process planning are described against the background of their usefulness for estimating direct manufacturing costs in detail design. Thus, there are a number of requirements to be met::

- Interpretation of the product model with respect to geometric and technological information and interaction between the objects that constitute the product model.
- Selection and instantiation of appropriate operations and necessary resources.
- Determination of the resource consumption of every operation.

Research on process planning has a long tradition at the Laboratory of Design, Production, and Management at the University of Twente in the Netherlands. Several research prototypes have been developed: CUBIC, ROUND, XPLANE, PART, and PART-S (Houten, 1991; Vin, 1994). PART (Planning of Activities, Resources, and Technology) is a generative CAPP system for small batches of prismatic parts. It derives manufacturing process plans and NC toolpaths automatically (Houten, 1991). It applies feature recognition and expert system technology. Feature recognition interprets B-Rep-based product and blank models and then generates sets of hierarchically ordered MFs. PART selects the machining methods and resources that are needed, e.g. machine tools, cutting tools, fixtures, sequences of operations in setups, and computes cutting conditions. Taken into consideration are the feature tolerances, feature orientations, and the axis configuration of machine tools. PART can be tailored to the needs of a given company. Furthermore, additional knowledge can be input in the systems in the form of a hierarchical set of rules. Process planning carried out against the background of the available capacity of the workshops. PART-S (Planning of Activities, Resources, and Technology – Sheet metal) is based on the same methodology and architecture as the PART system. Its application area is small batch part manufacturing of sheet metal components (Vin, 1994).

Ho (1997) characterizes a feature-based approach to automated process planning and NC machining that includes tool selection, feed and speed computation, operations reordering, and NC code generation. In this approach a feature represents a class of objects that can be machined using similar operations. The crucial component is to decompose features into sequences of machining operations, i.e. link product and process. The machining tool-driven feature decomposition breaks down a feature according to the capability of the available machining tools. Manufacturing knowledge has to be formalized for reuse, i.e. how a feature is manufactured needs to be predefined.

To avoid having to acquire and maintain the knowledge of the process planners, which is an extremely expensive process, Feller (1992) proposes a method for supported knowledge acquisition. Artificial neural networks (ANN) are deployed, with process planning divided into three parts: building combined operation sequences, allocating machines, and determining manufacturing times. A separate ANN trained with suitable data is utilized for each of these tasks. The ANNs are employed to allocate operations and resources and to calculate the manufacturing times. The acquisition of knowledge is simplified since unproven, incomplete, and contradictory information can be used and a task is solved without the need for an explicit strategy. Yet, this does not allow continuous manufacturing knowledge acquisition. To perform learning for process planning, Chang and Chang (2000) also apply ANNs in a CAPP system. In contrast to these two approaches, Ewert et al. (1995) and Dürr and Ewert (1998) automatically generate rules from a number of examples to generalize manufacturing knowledge.

Apart from the approaches set out in the above paragraph there are different CAPP research prototype systems which utilize mostly specific knowledge. Case-based techniques are applied to process specific knowledge, thus enabling learning. The methodology of case-based reasoning is described in more detail in chapter 3.3.

Some early research work in the field of applying case-based techniques for planning in mechanical engineering domains has been done by Paulokat and Wess (1994) and by Yang and Lu (1994). Paulokat and Wess (1994) illustrate how the case-based prototypical system CAPlan/CbC is deployed in the domain of manufacturing planning for rotationally symmetrical workpieces. Instead of generalized knowledge, specific knowledge is used. The system is prototypically implemented using CAPlan, a nonlinear, partial-order planner. CAPlan/CbC employs derivational analogy to replay past solutions (Muñoz-Avila and Weberskirch, 1996). The product description is not based on features; instead it is built up from geometric primitives such as cylinders, cones, and toroids.

Approaches that are different from those mentioned above with respect to a feature-based product description were taken by Marefat and Britanik (1997), Márkus et al. (1997), Váncza (1998), Gerken (2000), and Tiwari et al. (2001).

Marefat and Britanik (1997) apply case-based techniques for the process planning of prismatic features, e.g. holes, slots, and pockets. For every feature, previous sub-plans are retrieved, modified, and stored in an abstract case. Sub-plans are merged in the global process plan, applying a rule-based feature sequencing mechanism to secure the correct order. Previous sub-plans are retrieved on the basis of the type of feature, the physical dimensions, and tolerances.

Retrieval is dependent on proximity and not on manufacturing similarity. Process planning in (Marefat and Britanik, 1997) follows the five steps set out below:

- Sequence features based on interactions and accessibility.
- Determine for each feature machining processes.
- Compute a certainty value for the sub-plan.
- Select appropriate tools.
- Generate a global plan.

Márkus et al. (1997) present an approach for feature-based process planning by retrieving and adapting previous, part family-related plans. The notion of part similarity is based on the morphology of the precedence graphs over the parts' feature sets and includes both design and manufacturing aspects. Retrieving and adapting previous, part family-related plans applying precedence graphs necessitates that these graphs can be generated. For a set of selected parts, plans represented in precedence graphs are generated as per the generative approach of process planning (Márkus et al., 1997). Thus, rules and frames are applied. New parts are matched to the previous cases consisting of part and planning information. New plans are created through modifying or combining previous plans. Further developments apply CBR for controlling of inspection planning activities (Váncza, 1998).

Gerken (2000) presents an assisting system for feature-based process planning based on CBR. A multi-step iterative decision process has to be run through. Retrieval in the case-based process planning system in (Tiwari et al., 2001) is effected at the part, and not the feature, level. It considers solely exact matches; i.e. parts are said to match only if all their parameters fall within a certain range.

This section has shown that two completely different approaches can be found in the literature: applying domain knowledge as specific or as generalized knowledge.

2.5 Shortcomings and Potential

The concepts and tools for cost estimation described above show considerable shortcomings with respect to the objective of this thesis. The integration of cost estimation in the product development process and the possibility of design-concurrent usage are not solved satisfactorily. In contrast to generative-analytical models, statistical models and analogy models determine costs in a lump-sum fashion in the vast majority of cases. They are, in particular, not able to identify the cost-driving product characteristics of the products to be calculated because of an insufficient degree of detail and differentiation and thus do not permit cost-based comparisons between alternative products. Furthermore, no adequate concept for estimating purchased part costs as part of the production costs has been presented.

The absence of an explicit domain model which largely holds true for manufacturing in mechanical engineering, the difficulty to acquire the knowledge necessary to generate the domain model, and the effort involved in keeping it up to date have been insufficiently considered and remain to a great extent unsolved. This applies for the generative-analytical concepts and tools as well as for the concepts and tools for automatic process planning. For example, in rule-based systems, the acquisition and maintenance of domain knowledge are extremely difficult and may sometimes even be impossible. The application of CBR seems to be a workaround, but the CBR concepts elaborated for planning and cost estimation are still far away from applicability. In particular, the interpretation of the product model including the interdependencies and the retrieve and the reuse steps have not been solved sufficiently. How the similarity measure is defined has, for the most part, been neglected.

To conclude, none of the methods currently employed for the design-concurrent cost estimation of production costs is as effective and efficient as required. None of the cost estimation systems set out in this paper has found widespread application within industry. Although PART is commercially available, other automatic process planning systems have not achieved significant industrial use. A fully automatic process planning system still seems to be far away. Except for special applications with well-defined constraints, automated process planning is not applied. High efforts for acquisition, structuring, and formalization of knowledge prevent practical use. A number of issues have been left open, and case-based planning applications for manufacturing process planning are still far-removed from achieving deployment in real-world applications. However they do seem to be a promising approach for cost estimation in detail design.

This leads to the hypothesis of this research work. In the absence of an explicit and complete domain model together with the difficulties involved in acquiring and maintaining manufacturing knowledge, the estimation of direct manufacturing costs must mainly rely on the processing of specific manufacturing knowledge. CBR is the most promising methodology to tackle this issue but needs to be adapted to the requirements of cost estimation. The next chapter introduces knowledge and knowledge processing to enhance awareness of the difficulties involved in acquiring and maintaining knowledge and explains which characteristics of a domain indicate that CBR might be a suitable approach.

Chapter 3

Knowledge Processing

A great portion of the cost estimation model for direct manufacturing costs discussed in this work is related to knowledge and knowledge processing. For this reason, this chapter gives an overview of knowledge, knowledge processing, the methodology of CBR, and the characteristics of similarity and intends to sensitize the reader to the difficulties involved in acquiring and maintaining knowledge.

3.1 Knowledge

3.1.1 The Notion of Knowledge

According to Drucker (1993), we are entering the knowledge age, in which the only meaningful resource is no longer capital, nor natural resources, nor labor, but knowledge. Many sectors, e.g. manufacturing, will be based on knowledge, and business organizations will evolve into knowledge creators (Nonaka and Takeuchi, 1995).

Humans have knowledge, the ability to conclude, and the capability to learn. But, what exactly is knowledge? This is surely a non-trivial question as it has lain at the roots of philosophical investigations for over two millennia. The purpose of the remainder of this chapter is not to discuss or cover any particular detail and to give a generic definition of knowledge, but to address those issues relevant for a common understanding of this work.

Epistemology, a branch of philosophy concerned with the theory of knowledge, defines knowledge as adequate perception. Apart from this philosophical perspective, a short overview of definitions from two other scientific disciplines, psychology and artificial intelligence (AI), is set out below:

- In psychology, knowledge is the result of a process of cognition about conditions, i.e. cognitive units, and their properties and relations to other units (Häcker and Stapf, 1998).
- In AI, knowledge is seen as "the symbolic representation of aspects of some named universe of discourse" (Frost, 1986). Knowledge in knowledge-based systems (KBS) is understood as the codified experience of agents and experience as the source for problem-solving (Stefik, 1995). The term *codified* expresses that the knowledge has been formulated in such a way that it can be applied.

In everyday language, the terms *information* and *knowledge* are often used interchangeably, yet there is a clear distinction between them. Knowledge differs from information in the pragmatic dimension, i.e. knowledge has an intention and pursues an objective. This thesis understands data to contain a syntactic dimension, information to include semantics in addition to syntactics, and knowledge, additionally to the two former, to encompass a pragmatic dimension. Thus, knowledge is structured, embodied, relational, and context specific.

3.1.2 Tacit Knowledge

Over the past decade, theories and concepts for knowledge and experience capture and transfer have arisen in engineering sciences. In the epistemological dimension, the distinction between tacit and explicit knowledge (also embodied knowledge and theoretical knowledge) according to Polanyi (1966) has proved to be fruitful. Tacit knowledge is characteristic of the expert: it is personal, context-specific, and hard to formalize and communicate. Tacit knowledge incorporates cognitive and technical elements (Nonaka and Takeuchi, 1995). The cognitive elements center on mental models which are images and assumptions that humans have and in which they create artifacts by generating and manipulating analogies in their minds. They are the filter through which reality is cognized. The technical element of tacit knowledge includes concrete know-how and skills (Nonaka and Takeuchi, 1995). Explicit knowledge, in contrast, involves knowledge that can be articulated.

Tacit knowledge is inherent to the human domain expert and is, thus, of particular relevance for those interested in the process of KBS development. Knowledge engineering (KE) has emerged to describe this development process. It specifies the process of identifying, acquiring, formalizing, and implementing knowledge, irrespective of whether this knowledge is tacit or explicit. Hence, KE also deals with the issue of how to capture human heuristic and tacit knowledge. In process planning, for example, human experts enhance their skills through experience gained over the long term (Marefat and Britanik, 1997).

Nonaka and Takeuchi (1995) distinguish between four different methods of generating knowledge – socialization, externalization, internalization and combination – which all have their origin in the transition between tacit and explicit knowledge. Table 3.1 gives an overview.

transition from \ to	tacit knowledge	explicit knowledge
tacit knowledge	socialization	externalization
explicit knowledge	internalization	combination

Table 3.1: Methods of Generating Knowledge.

Nonaka and Takeuchi (1995) understand the transition from tacit to explicit knowledge as externalization. This is the most important method of knowledge generation for this thesis. In fact, it is the documentation and formalization of knowledge that has thus far existed only in the mind of the expert. The derivation of simple if-then rules and constraints, the acquisition of models and ontologies, and similarity judges are some examples of this area. Also, explicit knowledge can be utilized to combine different elements of explicit knowledge, thus extending the knowledge generally accessible.

There are two conceivable reasons for this:

- Even if experts are willing to reveal their knowledge, the difficulties are that they do not mention all the important facts because these are, in their opinion, self-evident.
- The expert adopts (socializes and internalizes) knowledge over a very long time.

Dreyfus and Dreyfus (1986) mention that experts are often not able to explicitly describe their knowledge because problem-solving is based not on using symbolic knowledge but on applying holistic models. The vast amount of knowledge is therefore not explicit but tacit in nature. This phenomenon is usually referred to as “the iceberg of knowledge”.

3.2 Knowledge Processing

3.2.1 Knowledge Acquisition

Many of the problems which are tackled by experts in everyday work are extremely complex. Engineering problems, in particular, are not amenable to purely algorithmic solutions and are usually ill structured (Sriram, 1997). A well-formulated procedure is often not available. Knowledge-based systems deal with difficult, ill-structured problems in complex domains (Sriram, 1997). A KBS is a model of a specific part of the real world that the system should be able to reason about (Aamodt, 1995) and incorporates and represents the experience and expertise of a human expert. But the question is what really makes a system knowledge-based? It is the presence of a knowledge base, a collection of explicitly represented facts about the world in combination with an inference mechanism, which is capable to reason about these facts (Levesque and Lakemeyer, 2000). Hence, KBSes represent not only the experts' knowledge but also their ability to draw conclusions (Puppe, 1991). Apart from the risk of oversimplification, a KBS can be divided into two main parts: the knowledge base, i.e. the repository for the knowledge utilized by the KBS, and the inference engine. Thus, a KBS separates the knowledge base from the control strategy.

Knowledge acquisition is usually the most difficult and most expensive process in developing a KBS, leading to it often being called the *knowledge acquisition bottleneck*. In general, knowledge acquisition is understood as the process of gathering knowledge and utilizing this knowledge to build a KBS (Smith, 1996; Sriram, 1997).

The elicitation of expert knowledge and its effective transfer to a useful knowledge-based system is complex and involves a diversity of activities. Initially, it was solely the transfer perspective that was considered to be relevant for knowledge acquisition. This is based on the fact that the expert's knowledge, which has to be represented, is explicitly available or can be extracted directly and can be subsequently stored in the knowledge base. This approach necessitates that the expert's ability to solve problems be primarily based on explicit and not on tacit knowledge. However, usually expert knowledge is not explicit. Therefore, knowledge acquisition is not just simply a transfer as there are inherent difficulties in making tacit knowledge – particularly in combination with everyday knowledge – explicit. This insight is in contradiction with the transfer view. The transfer view has proved to be erroneous. A paradigm shift from the transfer view to the construction view has occurred. The latter considers knowledge acquisition as a constructive process covering tasks such as domain analysis, knowledge elicitation, knowledge representation, implementation, and validation of the knowledge (Aamodt, 1995). Thus, the first step is to extract the knowledge from the expert's mind, which has already been referred to as externalization. Knowledge acquisition methodologies such as KADS (Knowledge Analysis and Design Support) (Schreiber et al., 1993), CommonKADS (Schreiber et al., 2001), and sloppy modeling (Morik et al., 1993) support the process of knowledge acquisition by providing a structured procedure.

Any knowledge acquisition task may be accomplished by manual methods (knowledge acquisition methods) or by automatic ones (machine learning methods) (Aamodt, 1995). Machine learning programs automate the knowledge acquisition process by providing a KBS with a learning capability. Learning systems are largely limited to analytical tasks, e.g. classification or diagnostics (Wess, 1996). To complete, cost estimation is a synthesis task.

3.2.2 Knowledge Representation and Reasoning

There are different methods for knowledge representation and reasoning for the various types of knowledge. These knowledge types do not necessarily cover disjunct areas. In the following, the emphasis is placed on the knowledge representation formalisms and inference mechanisms relevant for case-based KBSes.

As mentioned before, the two main parts of a KBS are the knowledge base as the repository for the domain knowledge and the inference engine, which utilizes procedural knowledge. The domain knowledge is declarative knowledge and contains a collection of facts about the domain of discourse, i.e. it describes the objects and their relations. In contrast, procedural knowledge is a method which is process oriented and specifies the procedure of how to derive a solution from declarative knowledge. It comprises a collection of instructions that offer support in searching a procedure for transferring the given initial state into the desired final state. Procedural knowledge is knowledge about how to do something. To solve a given problem, both declarative and procedural knowledge are necessary.

The knowledge base is maintained in order to keep up with the changes and extensions of the domain of discourse. The difference between KBS and traditional systems is that the latter do not separate domain knowledge and inference knowledge. They incorporate both in algorithms. If the domain knowledge changes, the knowledge base of a KBS can more easily be adapted or exchanged. Learning systems are able to automatically acquire knowledge and represent it in the knowledge base.

Knowledge modeling is the process of putting knowledge relevant to a certain domain into a predetermined form of knowledge representation. Knowledge representation is the way the knowledge is expressed. Knowledge representation does not only need to represent knowledge easily and express it clearly, it must also enable effective and efficient processing. It has to ensure aspects such as correctness, extendibility, and modifiability. Each of the various knowledge representations has its pros and cons and fits best for a particular type of knowledge or task. The applicability of learning methods, in particular, is driven by the type of knowledge representation.

In addition to the representation of knowledge, knowledge processing is of great significance. Reasoning is the formal manipulation of the symbols representing a collection of propositions (Levesque and Lakemeyer, 2000). Puppe et al. (2000) distinguish between two different problem-solving approaches: analytical and synthetic. Whereas the former selects the problem solution, the latter generates the problem solution using primitives. Basically, each solution of a technical task is based on searching. This applies for both the human expert and for computer systems.

3.2.3 Learning Capability

Learning encompasses a multitude of phenomena (Herrmann, 1997). The idea to develop learning systems is driven by the fact that maintenance of knowledge bases is extremely tedious and expensive. It is much easier to utilize existing data sets. The ability to learn is one of the central characteristics of intelligence (Langley, 1996). The question is: What is learning? Finding a satisfactory definition for this term is not necessarily any easier than exploring what knowledge is. According to Minsky (1986), "learning is making useful changes in the workings of our minds". And in artificial intelligence, learning can be defined as the ability to adapt to changes in the environment (Sriram, 1997). In general, learning is the process of gaining knowledge.

Bergmann (1996a) differentiates three types of learning:

- Analytical learning (deduction), e.g. explanation-based learning.
  (deductive conclusions, truth preservation guaranteed, conclusions prove correct)
- Synthetic learning (induction), e.g. inductive logical programming, decision trees.
  (inductive conclusions, truth preservation not guaranteed)
- Analogical learning (analogy), e.g. derivational and transformational analogy, case-based reasoning (CBR).
  (analogous conclusions, truth preservation not guaranteed)

Furthermore, supervised and unsupervised learning are distinguished. An important distinction is whether the learning phase and problem-solution phase run in parallel or not. For each of the three learning types mentioned above, Bergmann (1996a) has described the relationship between the confidence and the extensiveness of the knowledge learned. Figure 3.1 illustrates Bergmann's (1996a) conclusions. For comparison, the extensiveness for direct retrieval is very small while the confidence is at its peak.

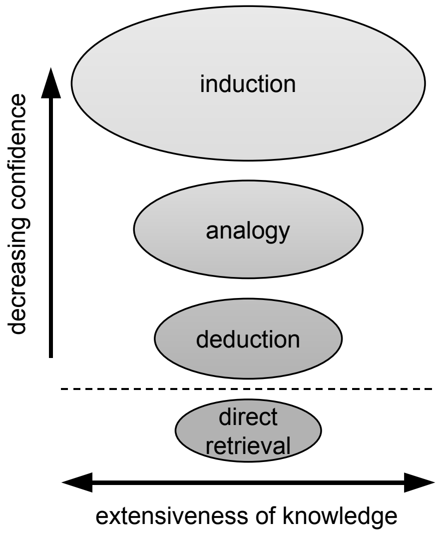

Figure 3.1: Confidence and Extensiveness of Knowledge Learned.

Analogical learning transforms existing knowledge for the fulfillment of a similar task. The difference between analogical learning and induction and deduction is that the former does not explicitly generalize the knowledge. Thus, learning by analogy is termed *lazy learning* in contrast to *eager learning*. Lazy learning algorithms differ from others in that they delay computation (Aha, 1998). For this reason, lazy learning stores specific experience and problem-solving is based on reuse of specific experience, i.e. interpretation of experience in contrast to compilation of experience as is done by induction. An example for analogical learning is case-based reasoning (CBR), which will be described in detail in the following section.

3.3 Methodology of CBR

3.3.1 Introduction to CBR

CBR serves as a methodology for developing knowledge-based systems. It is an approach geared for solving problems and for learning (Aamodt and Plaza, 1994). In particular, CBR stands for the way people solve problems by using cases as well as for the ways people can program machines to handle them (Kolodner, 1993). A CBR system solves new problems by adapting solutions of similar, previous problems (Riesbeck and Schank, 1989). CBR has emerged from research in cognitive psychology as a model of human memory to a methodology with increasing attention in theory and in practice.

Instead of pursuing a precise path that is easy to formalize when having to solve a new problem, humans usually remember a previous, similar problem and transfer the solution derived to solve the new problem. Should there be differences between the current and the previous problems, the human adapts the retrieved solution to better fit the current problem. Assuming, for example, that the engine of a car does not start, the experienced mechanic will remember previous cases in which the engines of other cars also failed to start. Running through the cases in his mind, the mechanic tries to figure out the most similar case by comparing similar boundary conditions or symptoms such as the battery voltage, the age of the car, and the behavior when the ignition key is turned. The previous solution is then reused in the current context. When transferring the retrieved solution, e.g. replace the old battery, the mechanic adapts the solution to the changed situation: for example, the type of the currently disabled car is different so a higher battery voltage will be needed.

Since human experts are not systems of rules but libraries of experiences (Riesbeck and Schank, 1989), the basic idea behind CBR reflects the way how human experts solve problems: using specific, episodic knowledge in the form of cases rather than using generalized knowledge. Thus, in CBR the primary knowledge source is not a set of generalized rules but a set of cases recording specific episodes (Leake, 1996). More information concerning the scientific background of cognition is cited in Kolodner (1993), Riesbeck and Schank (1989), and Schank (1982).

CBR is based on two tenets (Leake, 1996; Kolodner, 1993):

- Similar problems have similar solutions.
- Future problems are likely to be similar to current problems.

CBR implies the following procedure:

- Episodic knowledge – a problem description and its solution – is stored in the form of a case in the case base.
- To solve a new problem, the most similar past problem is retrieved.
- The past problem solution is reused in the new context.
- New experience is stored as a new case.

A case base is a collection of adequately organized cases (Bergmann, 1996b). Watson and Marir (1994) mention that cases can be represented in a variety of forms using the full range of AI representational formalisms, e.g. frames, objects, semantic nets, rules. Usually, a distinction is made between an episodic case, which is a case in a data resource and a prototypical case, which the user of a CBR application has explicitly defined. Figure 3.2 depicts this procedure and introduces similarity and utility.

The less a retrieved solution has to be adapted, the more useful it is in solving the current problem. As a result, the uncertainty about the quality of the solution decreases. In the absence of alternatives, in CBR the a posterio criterion of utility is reduced to the similarity of problem descriptions (Wess, 1996). The difficulty involved is a kind of paradox: because if two problems are similar, the solutions are similar. Furthermore, this statement implies that slight changes in the problem description will cause slight changes in the solution. Thus CBR cannot guarantee a 100% correct solution (Richter, 2000). In fact, exactness of solution is traded off for an approximate solution that is controlled by the similarity measure. Thus, according to Richter (2000) utility instead of truth is brought to the forefront.

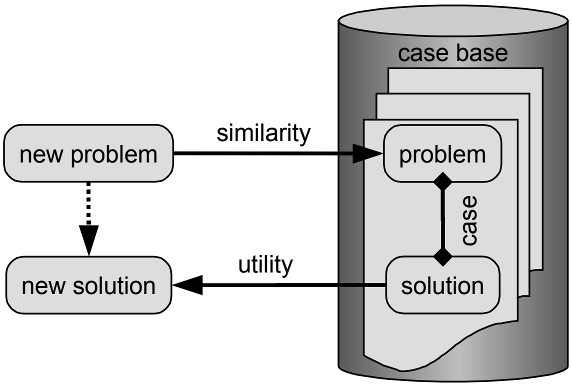

Figure 3.2: Similarity and Utility in CBR.

3.3.2 Process Model

For the process of CBR, literature proposes different process models. Generally, the process model according to the notions of Aamodt and Plaza (1994), which defines the case-based reasoning process as a cycle containing four sub-processes, is referenced. This process model identifies not only the key sub-processes of a CBR cycle but also the interdependencies and the products. Figure 3.3 depicts a redraw of this process model:

- **Retrieve** the most similar case or cases.
- **Reuse** the information and knowledge in that case to solve the problem.
- **Revise** the proposed solution.
- **Retain** the parts of this experience which are likely to be useful for future problem-solving.

Generalized knowledge, i.e. generalized domain-dependent knowledge as opposed to specific knowledge embodied by cases, plays a part in the CBR cycle by supporting the CBR processes. The support ranges from nil to very strong, depending on the type of CBR method employed (Aamodt and Plaza, 1994).

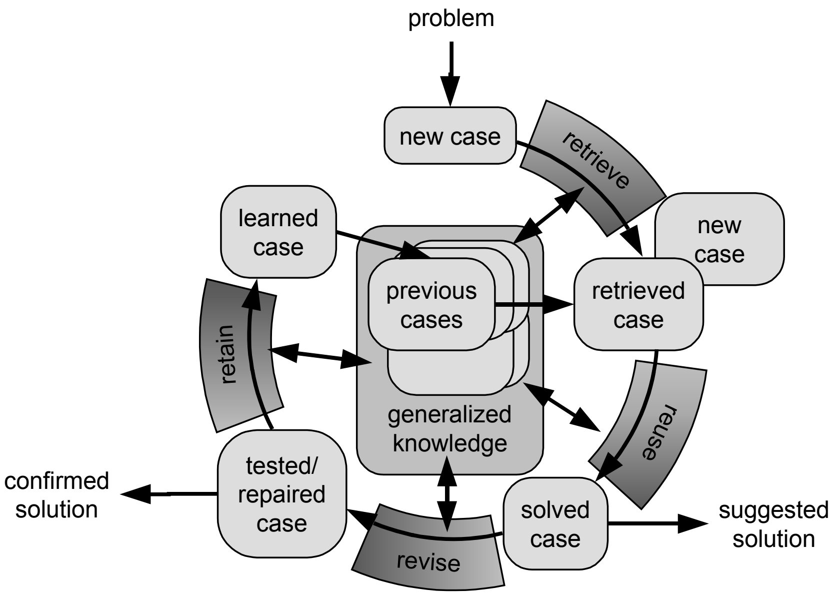

Figure 3.3: CBR Process Model.

Wess (1996) extends the model introduced by Aamodt and Plaza (1994), distinguishing explicitly between provision and selection of cases and between internal and external criticism of the retrieved and reused solution. And even this model comprised of six steps is still very general. In particular, each task is made up of a number of subtasks. Furthermore, each subtask can be implemented using a variety of techniques (Lenz et al., 1996). Since CBR does not prescribe any specific technology, Watson (1998) emphasizes that CBR is a methodology, not a technology. To go back to basics, the 4 Re's – retrieve, reuse, revise, and retain – will be described in more detail.

Case retrieval is the most crucial step in CBR as it directly determines effectiveness and efficiency of a CBR application (Mántaras and Plaza, 1997; Wess, 1996). Here similarities are used to call up a case or a number of cases which would seem to lead to a solution to the current problem (Bergmann, 1996b; Kolodner, 1993). Case retrieval usually necessitates a combination of search and (partial) matching (Kolodner, 1993). Belkin and Croft (1987) elaborate on retrieval techniques and subdivide them into exact-match and partial-match retrieval techniques.

A number of cases assumed to be relevant on the basis of their similarity are pre-selected. This is a search problem (Wess, 1996): the pre-selected cases are compared to the problem case using a more sophisticated matching procedure (Lenz et al., 1996). A case in the case base is not necessarily expected to match a new problem exactly. Thus, the search must result in the best partial match.

Reuse is concerned with generating a solution for the current problem by applying the knowledge that is contained in the retrieved cases (Bergmann, 1996b). The simplest way of reuse is to take the retrieved solution as the overall solution to the current problem. Wilke and Bergmann (1998) and Goos (1996) distinguish between CBR applications which merely copy the retrieved solution and adaptive CBR applications. The retrieved solution is modified or the solutions of multiple cases are combined to produce a new composite solution, which is referred to as compositional adaptation. Wess (1996) differentiates between complete and partial transfer as well as between singular and multiple transfer. Partial transfer means that only a portion of the retrieved solution is reused. Multiple transfer is similar to compositional adaptation, combining the solution of multiple solutions but without adaptation.

Because the match made is inexact, it is necessary to validate the solution suggested (Richter, 1998). The revise step checks the solution generated in the reuse step and, if the solution proves incorrect, modifies it (Bergmann, 1996b). Revision results in simulation or validation in the real world (Bergmann, 1996b). The latter provides an opportunity to learn from mistakes.

Retention is the process of incorporating what is useful to keep into the existing knowledge (Aamodt and Plaza, 1994) in order to enhance or sustain efficiency. Learning in CBR, which is driven by both the success or failure of the solution proposed (Leake, 1996). It is not necessarily restricted to lazy learning (retaining new and deleting unnecessary cases) but may also extend to updating of retrieval knowledge to enhance retrieval, e.g. weighting coefficients (Wettschereck and Aha, 1995), similarity functions (Craw et al., 2001), and adaptation knowledge (Jarmulak et al., 2001; Wilke et al., 1996) to enhance reuse.

The explanation of the CBR process steps illustrates that CBR applications require both episodic knowledge (interpreted at runtime) and generalized knowledge (compiled knowledge) (Richter and Althoff, 1999). Wess (1996) identifies two essential elements that necessitate domain-specific, generalized knowledge:

- The knowledge to retrieve and select the proper cases.
- The knowledge to adapt and modify the retrieved solution.

This kind of knowledge has to be acquired and represented in the CBR application in advance. Richter (1995a) introduced the notion of knowledge containers and introduces the following four containers in which CBR applications store knowledge:

- The vocabulary knowledge to describe the domain.
- The retrieval knowledge used to retrieve similar cases.
- The adaptation knowledge to transform and adapt solutions retrieved.
- The knowledge in the cases.

Knowledge in CBR applications is expressed using certain suitable types of knowledge representation.

3.3.3 Suitability

Case-based reasoning has become widely popular due to a number of advantages:

- CBR does not necessitate an explicit domain model; i.e. it is applicable for open domains.
- CBR is able to utilize available data sets as cases, thus promoting a reduction of knowledge acquisition effort.
- CBR does not require the domain experts to articulate their thought processes when solving problems.
- CBR allows the case base to be developed incrementally, while maintenance of the case library is relatively easy and can be carried out by domain experts. In contrast to this, rule bases are hard to maintain because of the interdependencies between the rules.
- CBR permits contradictions in the cases.

Nevertheless, CBR also suffers from a few drawbacks and Wess (1996) advises that the capability of CBR applications not be overestimated. Cunningham and Bonzano (1999) and Cunningham (1998) mention that, for constructing CBR applications, some types of problems still require significant knowledge engineering effort. Indeed CBR does not call for an explicit domain model but requires domain-specific knowledge for retrieval and reuse. Not only may this sometimes be as difficult to acquire as the domain model itself, it may even shift the knowledge acquisition bottleneck from the domain model to the knowledge concerning similarity.

Some of the characteristics of a domain which indicate that CBR might be a suitable approach are set out below:

- An explicit domain model is not available.
- Data sets of previously solved problems exist.
- Domain experts find it hard to articulate their thought processes when solving problems and talk about their domain by giving examples.

3.4 Characteristics of Similarity

In CBR, the retrieval of the most similar case is crucial as this substantially drives the quality of the solution. Hence, an adequate similarity measure is needed. Unfortunately, a generic similarity measure does not exist, because similarity always refers to a specified objective (Göker, 1998). Also, there is no universal theory of similarity assessment (Wess, 1996). Richter and Althoff (1999) state that "similarity is not an absolute notion ... but is always relative to some aspects". Therefore, additional, domain-specific knowledge is required to determine whether two different attribute values mean the same thing, something similar or something completely different (Goos, 1996). Thus, (background) knowledge drives the assessment of similarity.

Figure 3.4 depicts that similarity assessment is not possible without a specified objective. When assessing the similarity between the four objects with respect to size, the results would be *A* similar to *C* and *B* similar to *D*. Taking the shape of the objects into consideration yields *A* similar to *B* and *C* similar to *D*, while with the color the discriminating criterion *A* is judged similar to *D* and *B* similar to *C*.

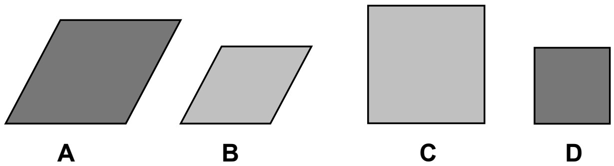

Figure 3.4: Assessment of Similarity.

Similarity possesses some decisive characteristics: Tversky (1977) states, in the contrast model, that the similarity between two objects is a function of their common and distinctive properties. Wess (1996) cites reflexivity, symmetry, transitivity, and monotonicity as characteristics of similarity assessment. Figure 3.5 depicts the example objects for these characteristics.

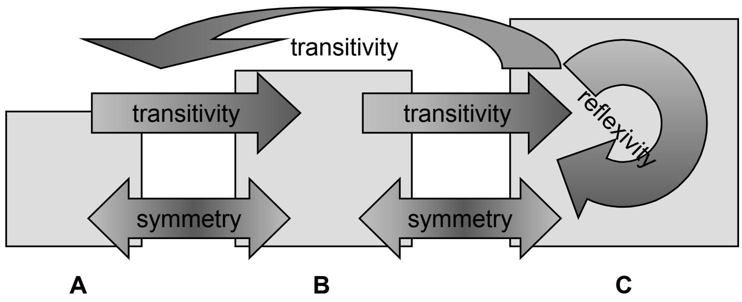

Figure 3.5: Characteristics of Similarity.

- Reflexivity: an object is similar to itself.
- Symmetry: if *A* is similar to *B* than *B* is also similar to *A*.
- Transitivity: if *A* is similar to *B* and *B* is similar to *C* then *A* is also similar to *C*.
- Monotonicity: the similarity between *A* and *B* grows monotonically with increasing similarities and decreasing dissimilarities.

Yet, especially the latter three do not necessarily always hold true. Similarity is sometimes accorded direction, so that the two objects *A* and *B* are not exchangeable and symmetry is therefore not always given. For a certain domain, e.g. manufacturing engineering, symmetry may largely hold true. Similarity can be formalized in three different fashions (Petersohn, 1997; Wess, 1996):

- Similarity as a predicate.
  This absolute notion only contains the binary information whether two objects *A* and *B* are similar or not.
- Similarity as a preference relation.
  This relative notion states that object *A* is more similar to *B* than to *C*.
- Similarity as a measure.
  In addition to ordinal information, this metrical notion quantifies the degree of similarity between objects *A* and *B*.

As first two formalizations can be derived from the third, this one is the most appropriate. In addition to the above alternatives for formalization of the notion of similarity, there are also various ways to represent similarity: a table, a taxonomy (Bergmann, 1998), an object-oriented representation (Bergmann and Stahl, 1998) or a similarity function to compute from scratch. The table is usually applied for nominal attributes and for discrete, numerical attributes. A table can also be applied if symmetry does not hold true. It is even applicable if reflexivity would not hold true. The representation of similarity in a taxonomy or in an object-oriented notion typically depicts an interclass similarity. A similarity function is best suited for continuous attributes with distance functions usually applied to compute the spatial distance between two points in an n-dimensional vector space (Wilson and Martinez, 1997) and similarity functions applied to assess the spatial distance with respect to the specified objective.

If the similarity between two objects, e.g two parts, or two DFs, is to be assessed, the specified objective has to be known. Balasubramanian and Herrmann (1999), for example, describe similarity assessment for a variant fixture planning approach which targets retrieving, for a new product design, a useful fixture from a given set of existing product models and related fixtures. The specified assessment objective in this approach is design similarity. If the goal in reengineering products is to increase the number of identical parts in different products, similarity between parts would also be assessed with respect to design, i.e. taking into account the geometric and technological properties of the part. But, if what is to be assessed is whether the two objects (parts or DFs) are produced in the same way, the specified objective for similarity assessment is no longer design but manufacturing similarity.

As has been mentioned, cost estimation using analogy reasons from geometrical similarity to similar costs, i.e. the specified objective for similarity assessment is design similarity. This is done in the absence of a similarity measure that specifies cost similarity and under the assumption that design-similar product models incur similar production costs.

But, if it is to be assessed whether for object B the same operations (having the same technological specifications) and the same resources would be applied as have been for object A, the specified objective has to be manufacturing similarity. Manufacturing similarity takes into account that two objects – even if judged design-dissimilar – can apply the same operations and resources. To give an example, the design similarity of the two blind holes as shown in figure 3.6 is 0.5 and the manufacturing similarity is 1. This means that the design of the blind holes is only slightly similar since the depth of hole is different whereas the manufacturing is identical, i.e. the operations and their technological specification and the resources applied are the same.

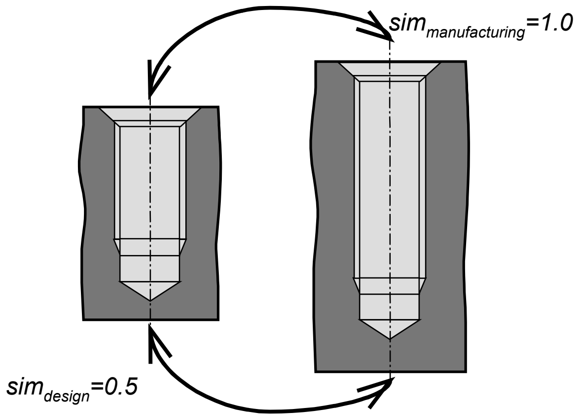

Figure 3.6: Assessing Design and Manufacturing Similarity.

PART II: Cost Estimation Concept

Chapter 4

Development of Cost Estimation Models with a Focus on Detail Design

4.1 Refining Requirements

The objective of this thesis is to elaborate a concept for concurrent estimation of production costs of discrete mechanical parts in detail design. In summary, the refined requirements to be fulfilled are as follows:

- Detailed and differentiated quantification of production costs.
- Adequate level of accuracy.
- Integration in the product development process.
- Interpretation of the CAD product model.
- Design-concurrent applicability.
- Cost-based comparison between alternative product designs.
- Applicability to the company's complete product portfolio (discrete mechanical parts).
- Ease of a priori domain knowledge acquisition.
- Continuous, ongoing domain knowledge maintenance.

Design-concurrent applicability means that a completely automatic derivation of cost estimate is irrefutable. The cost estimation report must be at hand without any delay. Knowledge acquisition tackles the problem caused by the lack of an explicit domain model and the difficulties met when having to acquire it. Knowledge maintenance includes refining, extending, and adapting the knowledge with the objective to become more experienced over time. Especially the two latter items in the list above are related to the hypothesis of this work: the estimation of direct manufacturing costs must mainly rely on the processing of specific manufacturing knowledge with CBR the most suitable methodology therefor.

Since, independent of any particular cost accounting model, production costs are made up of different cost elements with different characteristics, the decomposition into these elements facilitates the search for a satisfying solution.

Production costs comprise three main components:

- Direct costs for company-internal resource consumption.
- Direct costs for company-external activities.
- Indirect costs for company-internal resource consumption.

Production costs exclude product development and process planning costs. The direct costs for company-external activities are the monetary value charged for a product, e.g. rough part, or a service, e.g. external machining, i.e. it is the price paid. For costing based on indirect costs a certain inaccuracy and fuzziness is tolerated in favor of a reduced input effort. Cost accounting accumulates the indirect costs and assigns them to the cost objects by applying costing rates.

In conclusion, the detailed and differentiated quantification of production costs can be decomposed into three activities:

- Provision of prices to be paid for raw material or for rough parts and external services.
- Estimation of direct manufacturing costs.
- Computation of overheads.

Subsequently, the single costs for a current part can be aggregated in order to obtain the cost estimate for the production costs.

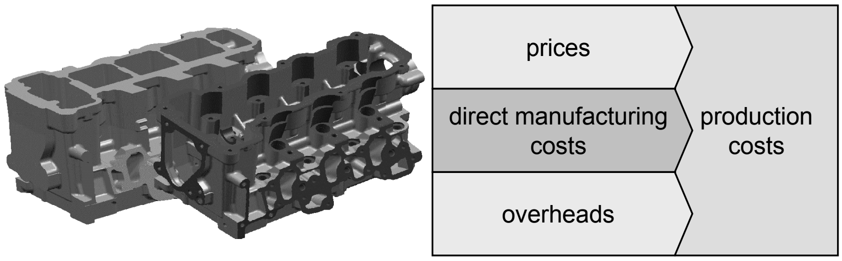

Figure 4.1: An Integrated Approach.

4.2 Prices for Company-External Activities

This chapter targets providing the prices to be paid for company-external activities as part of the production costs, i.e. prices for raw materials, rough parts, and external services. The specification of the unfinished part, i.e. determining the type (semi-manufactured product, cast, forged or sintered blank, etc.) and dimensions with regard to technological, economic, and temporal constraints and requirements, is usually done during product development. This thesis assumes that if a physical rough part exists, a CAD product model of the rough part is also available.

Since the focus of this thesis is on discrete, mechanical parts and not on assemblies, a supply chain of at most two stages is considered. The supply chain is part of the supply net as depicted in figure 4.2. Service providers perform value-adding, outsourced activities on the part. Outsourcing activities may be desirable for many reasons, e.g. cost-efficiency, low competence in an area, lack of capacity. This thesis excludes the scenario that activities are outsourced in the short run because of a capacity bottleneck or that they have been outsourced for the first time. Rather it is assumed that activities are outsourced in the long run for reasons of cost efficiency or lack of expertise. Furthermore, prices to be paid that are part of costing rates are not considered. The activities provided by self-sufficient profit centers inside a company that compete with the external companies for jobs can also be treated as company-external activities.

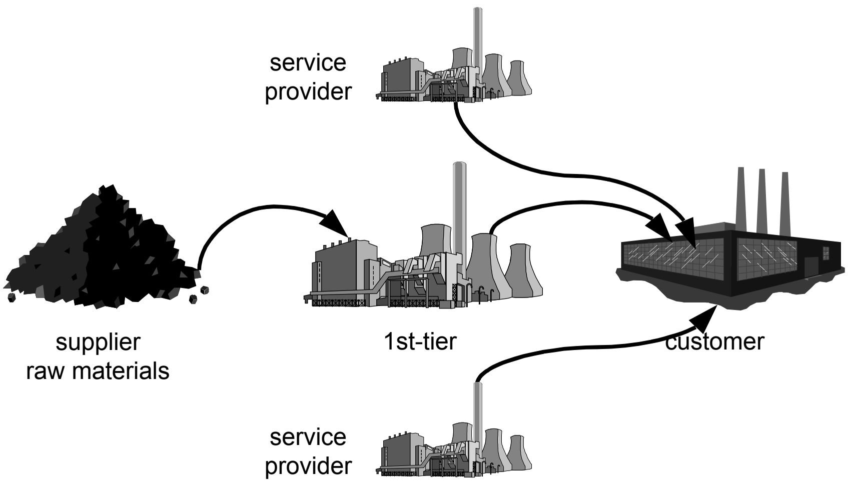

Figure 4.2: Supply Net.

In the supply net, there are a multitude of factors that impact the price to be paid for company-external activities. The most crucial are noted below:

- The supplier's company philosophy (pricing policy, strategic or operative objective, profit ratio, competition, etc.).
- The customer's supplier policy (short-term or long-term assignment, quality, punctual delivery, know-how drain, etc.).
- Variations in the global market prices for raw materials.

Thus, in addition to the geometric-technological product specification, these factors exert a grave impact on the prices to be paid. The estimation of these prices suffers from various drawbacks. The most crucial one is that the prices are driven by market demand. Moreover, the production process cannot be anticipated. Suppliers usually do not disclose their processes and costing rates. Neither is the targeted profit disclosed. Thus, estimating the total product costs of the services requested based on the anticipation of the production process is not viable.

The result of an exploratory study as part of a company-internal research project has shown that a very detailed determination of the prices to be paid for company-external activities can only be done in close collaboration with the purchasing department. Since such cooperation leads to a time delay and significantly expands the design-concurrent applicability, the cost estimation system has to be able to roughly estimate the prices. This thesis is not concerned with elaborating an approach that takes the soft facts and the insecurity mentioned above into account, rather it proposes that the data arising in Purchasing should be collected and prepared in order to be able to use it in cost estimation. A database is most adequate for collecting data related to prices for company-external activities.

Figure 4.3 offers an overview of the content of such a database. The database is subdivided into four categories: raw material, rough part, service, and general information. For the estimation of the prices of raw materials both the world market prices for various raw materials and the prices offered by the raw material suppliers are available. However, for the estimation of prices for rough parts, the prices paid and the prices quoted in the past are stored together with a description of the rough part. If the current and the previous rough part are similar with respect to manufacturing, either the price of the bid or the price paid is taken over. In the absence of knowledge about company-external manufacturing processes instead of manufacturing similarity design similarity will be applied. For external services the database again includes the prices paid and the prices offered. In addition, there is a correction factor to derive the price to be paid from an estimation of the cost for a certain activity based on internal processes and cost rates. This, of course, necessitates that the manufacturing knowledge about these processes be on hand, which unfortunately does not always hold true. Finally, the database provides the relevant rate of price increases in order to reason what the price would be today on the basis of past prices.

Figure 4.3: Database with Prices for External Activities.

4.3 Direct Manufacturing Costs

The most detailed and differentiated cost structure is obtained when the future manufacturing process is accurately anticipated and the resource consumption is assigned a monetary value. Hence, to estimate the direct manufacturing costs this thesis is concerned with elaborating an approach for anticipating the applied manufacturing processes.

Detached from real process planning activities, a provisional sequence of operations is generated, taking the CAD product models of the rough part and the finished part into consideration. Production scheduling and aspects which are solely of interest in process planning but have no impact on the manufacturing costs are neglected. This sequence comprises all the operations and resources needed to manufacture the part including their quantified resource consumption. Resource consumption is then given a monetary value with the specific cost rates of the allocated resources. The single costs are aggregated in order to obtain the total direct manufacturing costs and the detailed cost structure.

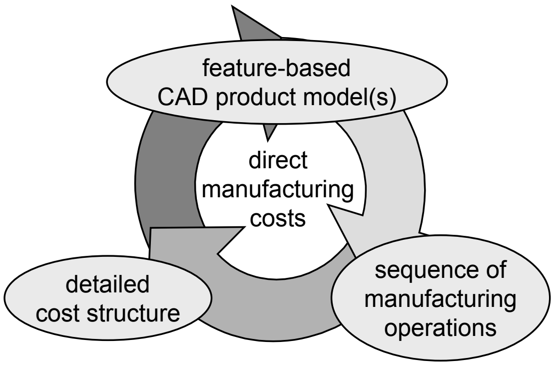

Figure 4.4: Approach for Direct Manufacturing Costs.

In general, generating a process plan is not a trivial task due to the inherent dependencies. In addition, the planner has to cope with the combinatorial explosion of possible solutions. Usually process planners apply an enormous amount of tacit knowledge and only a small portion of explicit knowledge. As described in chapter 3.1.2, it is extremely difficult and expensive – albeit sometimes impossible – to make tacit knowledge explicit. Furthermore, manufacturing knowledge is subject to a continuous process of evolution because of new materials, innovative manufacturing methods, etc. This necessitate that manufacturing knowledge can be modified or completely new knowledge be acquired and applied. To deduce a sequence of reasonable operations from the CAD product models requires manufacturing knowledge. This domain knowledge is largely tacit with explicit knowledge making up only the tip of the knowledge iceberg.

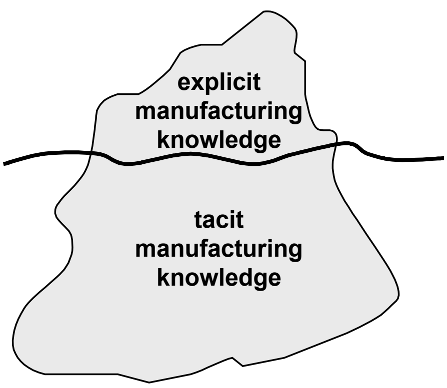

Figure 4.5: Iceberg of Manufacturing Knowledge.

This thesis is not concerned with the conversion of tacit knowledge into explicit knowledge, as this has been tried sufficiently by others in the past. Instead, this thesis captures and applies only what is already explicit, easy to make explicit or worth making explicit and searches for new paths in order to utilize the tacit knowledge. This also includes the ability to participate in the continuous evolution of tacit knowledge and manufacturing technology.

Manufacturing methods which have only few degrees of freedom are less complex. As a rule there is a simple relationship between the product characteristics and resource consumption. The manufacturing knowledge for these manufacturing methods is either mostly explicit or easy to make explicit. It is represented as generalized knowledge in the knowledge base for cost estimation.

In contrast to this, the knowledge about manufacturing methods which lead to a combinatorial explosion of possible solutions is typically not explicit nor is it easy to make explicit. Chapter 2.2 has described how features in product development and process planning are instantiated and how a persistent link between the instances is maintained. Thus, in their everyday work, process planners store their tacit knowledge in integrated PPR instance models. Despite the fact that these models are traditionally viewed as mere data rather than as knowledge, they implicitly contain manufacturing knowledge. The reason for this is that the knowledge is stored in such a way that it cannot be directly used. The manufacturing knowledge is present in terms of cases, i.e. DFs, MFs, resources, and their relations. Since CBR is able to apply cases for reasoning, the models as concrete descriptions of past scenarios become episodic knowledge. In order to be able to reuse this episodic knowledge of integrated PPR models for the initial retrieval step, knowledge concerning manufacturing similarity has to be acquired. It is beneficial that the acquisition of knowledge about manufacturing similarity is easier and less expensive than trying to fully acquire the tacit knowledge. Thus, the data of operative systems is applied for the informational cost estimation system in order to enhance decision support.

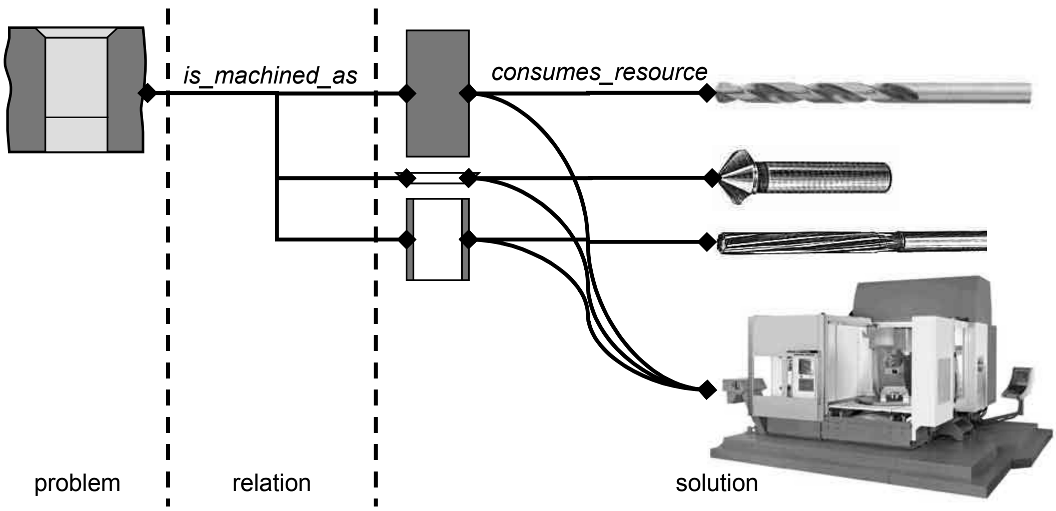

Figure 4.6: Data Set.

The methodology of CBR possesses some decisive advantages compared to other approaches such as rule induction or artificial neural networks. These are the ability to explore existing data sets, the ability to rely on episodic rather than generalized knowledge, and the ability to acquire new knowledge in terms of cases. Chapter 3.3 has provided a general description of when CBR is a suitable methodology. As these criteria still hold true, CBR is applied for the retrieval and reuse of past episodic manufacturing knowledge captured and exploited in integrated PPR instance models which cannot be effectively and efficiently captured in any other way. Furthermore, CBR is applied to retrieve and reuse specific knowledge explicitly defined and captured in prototypical cases. It is usually more intuitive for the domain expert to define prototypical cases than representing the knowledge in if-then rules.

It is the first two steps in the process model suggested by Aamodt and Plaza (1995) – retrieve and reuse – that are important for the interpretation of the data sets with the objective of applying the specific manufacturing knowledge. All the elements that impact manufacturing have to be taken into consideration for retrieval, except for those elements for which generalized manufacturing knowledge is represented. The information contained in the retrieved solution is reused in the current context, i.e. resource consumption is newly computed. The revise step is part of the reuse step and includes checking the plausibility of a generated solution.

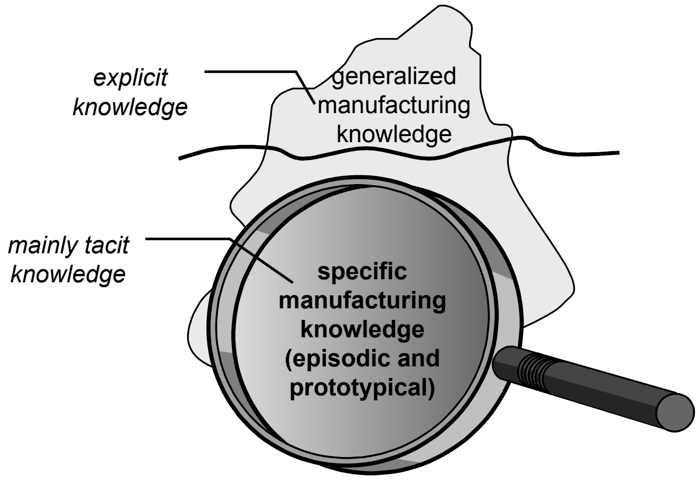

Figure 4.7: Hybrid Approach Based on Generalized and Specific Domain Knowledge.

4.4 Overhead Costs

This section illustrates how overhead is computed. Since many companies continue to rely on traditional cost accounting, this thesis also refers to the traditional cost accounting methods: the intention is not to improve cost accounting itself but rather to provide a model which supports the provision of cost information during the design process. In this regard, the cost estimation model does not seek to overcome the inaccuracies of traditional cost accounting – rather it looks for conformance.

As depicted in chapter 2.3.2, cost accounting aggregates costs at different levels. Production costs comprise all the costs occurring in manufacturing, including overheads. Calculation of overheads necessitates a basis to which they refer and an overhead rate. Bases for calculation of material overheads are, for example, the direct material costs.

Thus, applying differentiated job order costing based on machine hourly rates, which still enjoys widespread application within industry, the material overheads and the manufacturing overheads are computed as follows:

$$material\ overheads = direct\ material\ costs \cdot material\ overhead\ rate \tag{1}$$

$$manufacturing\ overheads = direct\ manufacturing\ costs \cdot manufacturing\ overhead\ rate \tag{2}$$

The material overhead rate and the manufacturing overhead rate are pre-calculated by Controlling and are fixed for a certain period, i.e. for the computation of the material overheads and the manufacturing overheads, they are independent of the impact of the design alternative's production. Thus, inherent dependencies are excluded. The direct material costs correspond to the total amount of prices to be paid for external activities related to one part. For this work, costing based on differentiated job order costing on the basis of machine hourly rates is not restricted to the monetary value allocated to the resource consumption of equipment but also includes costs accruing due to supporting and auxiliary supply. The cost estimation model is not confined solely to this type of cost accounting; other methods can be deployed or the costs (actual, normal, and budgeted) can be varied.

In conclusion, the production costs for this thesis are computed as the sum of four elements:

- An aggregation of the prices to be paid for company-external activities (i.e. direct materials in cost accounting).
- Material overheads.
- Direct manufacturing costs.
- Manufacturing overheads.

In the following chapter, solely the estimation of direct manufacturing costs will be addressed in further detail.

Chapter 5

A Closer Look at the Cost Estimation Model for Direct Manufacturing Costs

5.1 Partial Models and their Integration

The cost estimation model for direct manufacturing costs presented in this thesis is based on the integration of information of product, process and resource in one cost view, for each of which a partial model is developed. Chapter 5.9 provides an overview of the cost estimation model for direct manufacturing costs and how the modules developed interact.

A partial model can either be a class model, an instance model or both. Abstract classes cannot be instantiated but may be super-classes of instantiable sub-classes. The sub-classes inherit the parameters of their super-classes according to the object-oriented paradigm. Integrated models emphasize the integration of partial models, which means integration on the class level as well as on the instance level. Figure 5.1 shows the three partial models product, process, and resource, and their integration as well as specialization and instantiation relations in each partial model. The horizontal arrows represent integration relations (IR), the vertical arrows represent specialization and instantiation relations. The figure below further distinguishes between abstract and instantiable classes.

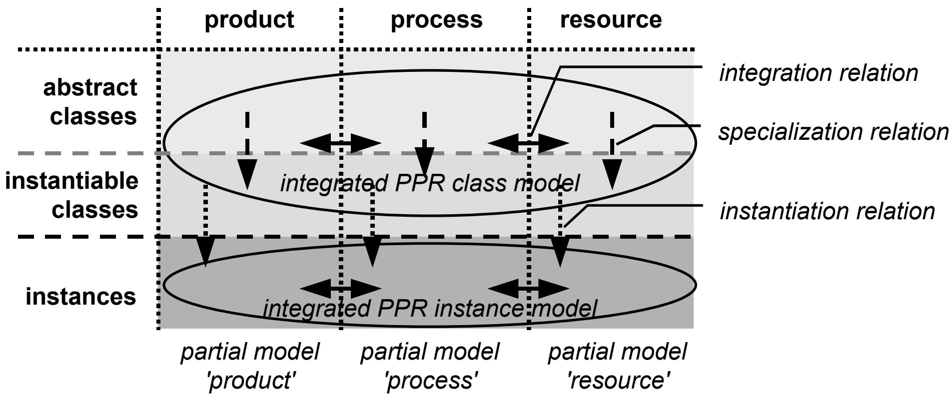

Figure 5.1: Integration, Specialization, and Instantiation.

The information that describes a product instance model has to be complete with respect to the application. For cost estimation in mechanical engineering this means that a CAD product model (product instance model) has to comprise geometric and technological information including specification of dispersion around the optimum where the function of the workpiece is still satisfactory. Moreover, since a CAD product model is built up of a large number of interacting objects, the relations between them are also crucial.

The product class model constitutes the building blocks and the blueprints to construct this instance model. While the specialization relation is employed to inherit properties, the aggregation relation (not shown in figure 5.1) is used to represent the part-of relationship. Interaction relations, e.g. intersection, describe the interaction between objects of the product model. The two instance models *process* and *resource* comprise the constituents of a complete process and resource description from the viewpoint of cost estimation. Again, the class models represent the repository to build up these instance models. Integration relations combine objects either on the class or the instance level. Both abstract and instantiable classes can be related to each other. In principle, the arity of relations on the class and instance levels is of type *m:n*, i.e. *m* objects of the product partial model are related to *n* objects of the process partial model, and the latter are again related to objects of the resource partial model.

According to ULEO (Zimmermann et al., 2002a,b), all types of relationships are defined in a taxonomy of relation types and are instantiated in the PPR model. For this reason, the relation types to be applied in the PPR model are not limited to a predefined number: they can be extended. For the integration relations a distinction is made between informational IRs and generative IRs. Informational IRs describe logical integration relationships between the objects, while generative IRs comprise the complete knowledge for automatically instantiating the related objects and their relation. This is similar to IEORs and GEORs in ULEO (Zimmermann et al., 2002a,b).

5.2 Product Model

5.2.1 CAD Product Model

As stated above, a CAD product model has to comprise geometric and technological information. In particular, geometric information is inherent to 3D CAD product models, whereas technology specification is not. This has to be explicitly defined and assigned to the workpiece itself or to an object of the workpiece to which it refers.

State-of-the-art high-end CAD systems support parametric modeling. Although feature-based design is thus far not generally applied and not yet all of its potential is taken advantage of, it is an underlying premise of this work. Feature-based design is not limited to the generation of pure geometry; it includes specification of the technology. In feature-based design, DFs and geometric elements (GE) are applied. A GE is a parametric, geometric primitive, e.g. body, face, edge, or vertex. Its shape and position are specified by parameters. A GE is not a functional element, i.e. an object with real semantics, whereas a DF is.

A DF is either a form DF or a technological DF. The former is made up of parameters and GEs, the latter is built up of parameters which drive technology. In the CAD product model, technological DFs are assigned either directly to the part or to a GE, which again can be part of a form DF. In addition to the DFs, a part has functional (material, tolerancing principle, etc.) and organizational (name, etc.) parameters.

In order to obtain comparable CAD product models, not only geometry but also a technology modeling method have to be observed. This includes three activities:

- Representation and application of DF classes.
- Adherence to a DF instantiation sequence.
- Observance of a tolerancing concept.

DFs represented in the product class model are applied to build the CAD product model. Furthermore, as figure 5.2 depicts, using the example of a pocket and a blind hole, this also necessitates that the form DFs be instantiated in the correct sequence. Independent from the sequence of the applied manufacturing operations (drilling and milling), these two form DFs have to be uniquely instantiated in order to obtain comparable form DF parameters. In the example set out below, the insertion plane of the blind hole is the bottom of the pocket. Adhering to this instantiation sequence, the blind hole becomes the child of the parent element *pocket*. As will be shown later on, the feed path for drilling is contingent on the sequence in which the manufacturing operations are carried out and can be computed for both drilling with subsequent milling and vice versa based on this DF instantiation sequence prescription.

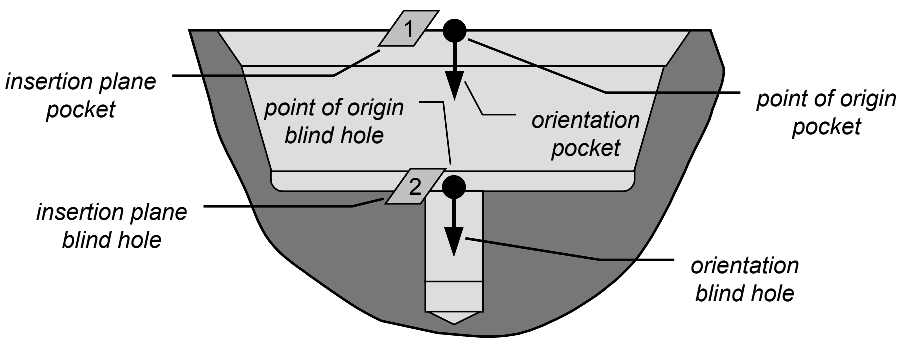

Figure 5.2: Form DF Instantiation Sequence.

In addition, the GD&T (geometric dimensioning and tolerancing) tolerancing concept, which applies ISO standards, has to be chosen (see appendix A.1).

The instantiation sequence can be different from the one described above; the same applies for the tolerancing concept. Yet, once a common DF instantiation sequence and a common tolerancing concept have been agreed upon, they have to be adhered to. This enhances the quality of the CAD product model data and facilitates its application in concurrent and subsequent engineering disciplines including cost estimation.

This thesis presupposes that an unambiguous digital model of both the rough and the finished part are available; furthermore it is assumed that these models are geometrically and technologically correct (material, tolerancing principles, dimensional and geometric tolerances, heat treatment, surface characteristics, etc.) and completely specified, based on a common modeling method.

5.2.2 Form DF

DFs as solution patterns for product development simplify, accelerate and standardize the design and modeling processes and add semantics to the product model. Form DFs are parameterized objects that consist of a topology of GEs and build the shape of the product model. Form DFs are usually applied in detail design. For further specification of the form DF, technological DFs can be related to the DFs or to the GEs which make up a form DF. Technological DFs are described in the next chapter.

For form DFs a distinction is made between depression and protrusion DFs. If a depression DF is instantiated in the CAD product model, the volume of the model decreases. In contrast to this, a protrusion DF increases the volume. To conclude by analogy that depression DFs are manufactured using material removal is false. A pre-cast through hole in the rough part model and through hole in the finished part model are examples for depression DFs. A cast boss and a rib are examples for protrusion DFs. Modeling the rough part usually calls for different form DFs than modeling the finished part.

Figure 5.3 shows the form DF *through hole*, which is standardized in DIN 32869-3–2002. Instantiating the through hole in the CAD model requires that geometric parameters such as diameter, depth, etc. as well as the point of origin and the orientation be specified.

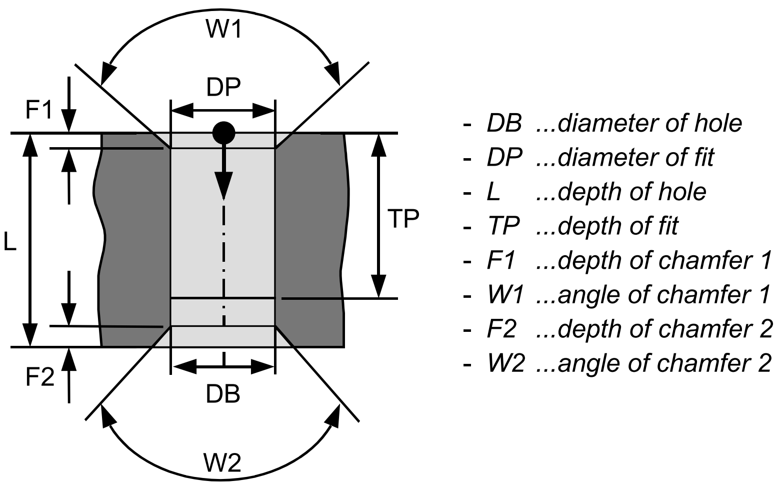

Figure 5.3: Through Hole DIN_32869-3-3 (DIN 32869-3–2002).

To give another example, Figure 5.4 shows the form DF *rectangular pocket* in the side view and in the top view. Instantiating the pocket means assigning values to the geometric parameters length, width, depth etc. as well as recording the point of origin and the orientation.

The absolute position of a form DF is marked by the point of origin and the orientation. For cost estimation, the orientation is needed to group operations with similar or identical approach direction into groups of sub-jobs. The point of origin is applied to compute the resource consumption for the positioning movement of the machining tool.

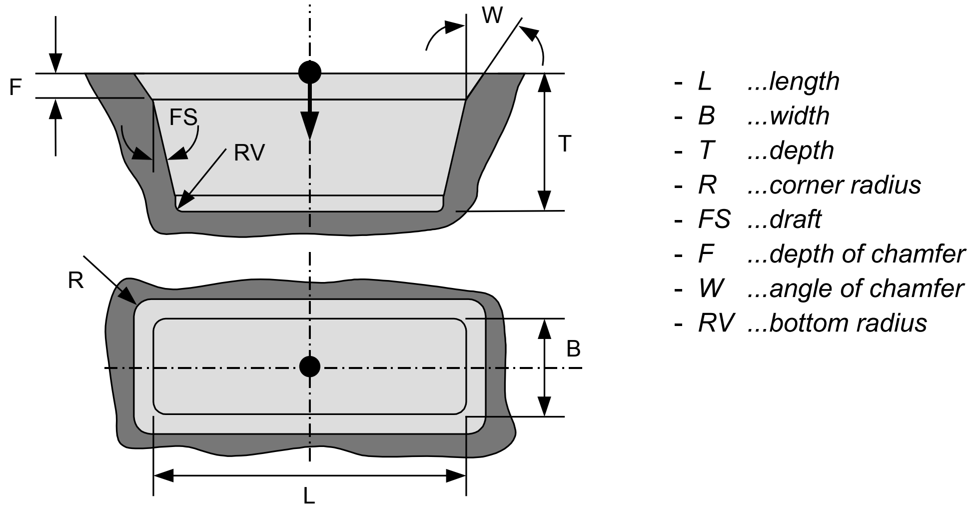

Figure 5.4: Rectangular Pocket DIN_32869-3-17 (DIN 32869-3–2002).

In addition to these absolute notions, for purposes of similarity assessment the relations between adjacent features are of considerably greater importance, e.g. the interaction of a form DF with adjacent form DFs. Examples for interaction relations are parent-child relations and intersection relations.

The example in Figure 5.2 depicts a parent-child relation between a rectangular pocket and a blind hole. The blind hole is nested in the pocket and is thus the child to the parent pocket. As set out, the modeling method prescribes the instantiation sequence. The bottom of the pocket is the insertion plane for the blind hole. Furthermore, parent-child relations with more than one child element are possible, for example two blind holes may be nested in the pocket. Or a pocket may be nested in a pocket which again is nested in a third pocket. Regarding the instantiation sequence described, the two form DFs do not intersect. In contrast, figure 5.5 shows two intersecting slots as another specialization of an interaction relation. An interaction relation describes the interaction of objects of the product model with respect to manufacturing. The intersection relation is further specified by the angle (here 90° for example) at which the two objects intersect.

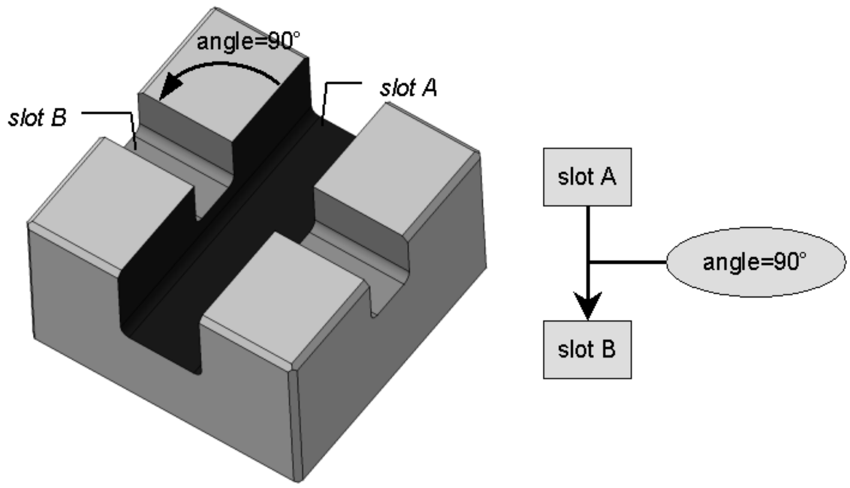

Figure 5.5: Intersection Relation.

These relations result in a net-like structure of the interacting elements. This is crucial for similarity assessment, which, in the first step, does not consider the elements to be independent from each other.

Even though this chapter has only treated examples of form DFs, the characteristics described apply equally to GEs. In particular, GEs and form DFs may also have interaction relations.

5.2.3 Technological DF

Technological DFs are manifested by the design engineer in order to ensure that certain technical demands are fulfilled: for example, to limit the deviations in the coarse (geometric tolerance) and in the fine structures (surface texture) since it is impossible to manufacture parts without any deviation from the nominal shape and ideal surface. Technological DFs are either valid for the complete part or solely for form DFs or GEs, with local requirements given priority over global requirements (DIN 32869-2–2002). Technological DFs generally result in additional, cost-driving processes or more expensive manufacturing methods. Some examples for technological DFs are cited below:

- Geometric tolerance DFs.
- Surface texture DFs.
- Heat treatment DFs.
- Surface coating DFs.
- General tolerance DFs.

For mechanical parts produced by machining, ISO 2768-1–1989 and ISO 2768-2–1989 describe the indication of general tolerances in drawings. The indication of heat treatment of ferrous metals is set out in DIN 6773–2001 and the indication of surface texture in ISO 1302–2002. In the absence of a technological DF standard, the specification of technological DFs is largely in conformance with the corresponding standards, which prescribe the way technology is indicated in drawings. For further information concerning geometric and general tolerances and surface texture, see appendix A.

Within the framework of this thesis, a surface texture DF in accordance with ISO 1302–2002 comprises the type of manufacturing process (removal of material required, removal of material not permitted), the specific manufacturing process (milling, grinding, etc.), one to *n* surface texture requirements, and the surface texture lay (parallel or perpendicular to plane of projection, crossed in two oblique directions, etc.). The surface texture requirement itself is made up of further specifications.

For geometric tolerance DFs as per ISO 1101–1983, a distinction is made between form tolerance DFs and tolerance-of-position DFs, with the latter again differentiated into orientation tolerance DFs and location tolerance DFs. Parallelism tolerance DFs and perpendicularity tolerance DFs are two examples of instantiable orientation tolerance DF classes. Position tolerance DFs and concentricity tolerance DFs are examples for instantiable classes of the abstract class location tolerance DFs. To round off the examples, flatness tolerance DFs and cylindricity tolerance DFs are instantiable form tolerance DF classes. If other standards than ISO 1101–1983 were to be applied, tolerance DF classes might be different. For ASME, as per ASME Y14.5M–1994, however it would be nearly the same.

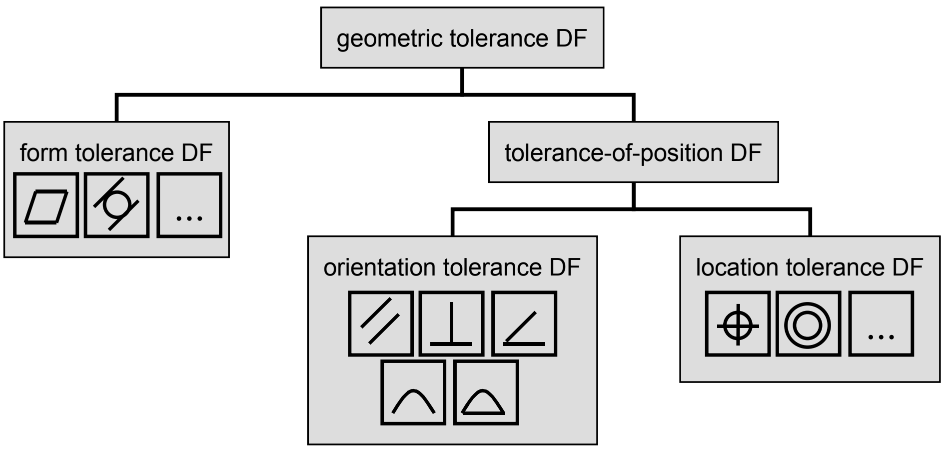

Figure 5.6: Tolerance DFs.

Flatness tolerance in ISO 1101–1983 is described by a tolerance zone which is limited by two parallel planes, the distance of which is the distance of the tolerance zone. Additionally, the maximum material requirement may apply (ISO 2692–1988). Thus, the flatness tolerance DF is specified by the following parameters:

- Identifier (e.g. flatness tolerance DF ISO 1101–1983).
- Distance of tolerance zone.
- Maximum material requirement.

The first parameter is a nominal attribute, the second a numerical one, and the latter is of type binary. For the cylindricity tolerance DF, the parameters are identical.

In contrast to form tolerance DFs, tolerance-of-position DFs refer to a datum, which consists of datum elements. A datum element is either a GE or GE of a form DF. For cost estimation it is of importance if the datum elements are GEs of the same compound as the toleranced element since this impacts the sequence of operations.

5.2.4 Product Reference Model

A reference model is applied to generically describe the structure of a class model, applying solely abstract classes and mainly abstract relations. The product reference model is the abstract model that sets out the structure of the product class model, which is usually not generic.

The product reference model incorporates the objects making up the CAD product model and their relations described above. The intention of the product reference model is to describe the product data independent from any particular system. The main components of the product reference model are the abstract classes product, DF, and GE. Each of these three classes can be further specialized. The objects of the product reference model have aggregation and interaction relations. From these, the product class model for a certain domain is constructed.

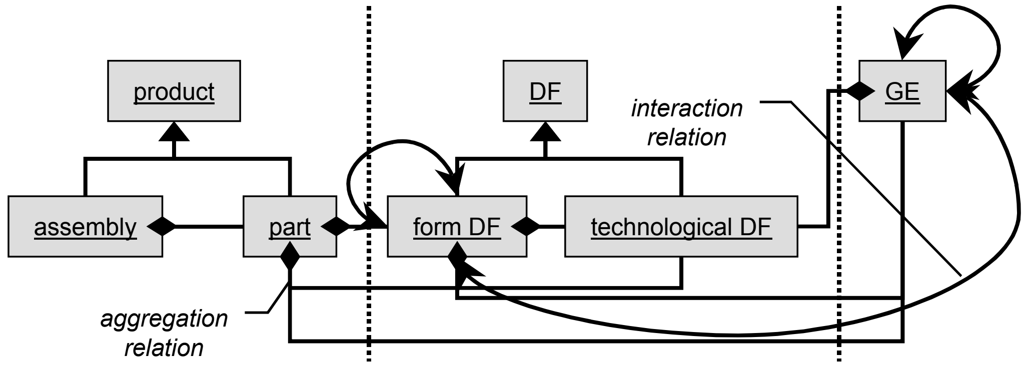

Figure 5.7: Product Reference Model.

Figure 5.7 depicts the product reference model. A product is either an assembly or a part. An assembly, which is not further pursued in this thesis, consists of parts. To sum up the relationships, a DF is either a form or a technological DF. GEs are applied to build form DFs or parts. A GE or a form DF is further specified by a technological DF, which can be also directly applied to the part. GEs and form DFs interact with themselves and with each other.

The product class model that is derived out of the product reference model consists of taxonomies of product types, DFs and GEs. The former are – as will be shown later in this work – mainly used to allocate operations to product types. The DF and the GE classes are the libraries which are applied to construct the CAD product model. The specialization relations in this object-oriented structure allow sub-classes to inherit the characteristics from the super-class. As mentioned previously, not only is it the objects of the product class model that can be specialized but also the relations.

5.3 Process Model

5.3.1 CAx Process Model

In this thesis, a process is understood as a production process, i.e. in manufacturing. A manufacturing process converts raw material or an unfinished part into a finished part. It is a systematized series of distinct operations. An operation itself is a discrete action performed to produce a desired (intermediate) result and pertains to a process. An operation is processed by one or multiple resources.

An operation in the library of the CAx system that serves as solution pattern is called manufacturing feature or inspection feature. The latter does not depend on whether the operation changes the geometric shape or the technological properties of part or whether it is a non value-adding activity. Thus, the terms operation and MF or IF are synonymously applied.

Figure 5.8 depicts a drilling MF. To compute the resource consumption for cost estimation the drilling MF is sufficiently described by the geometric parameters *approach clearance*, *breakthrough*, and *drilling depth* and by the technological parameters *machining* and *retract feed rate*. A geometric parameter of the MF corresponds either to geometric parameters of the form DF, e.g. drilling depth, to geometric parameters of the resource, e.g. nominal diameter of milling cutter in pocketing, or have no correspondence in the CAD product model, e.g. approach clearance. The latter are geometric parameters mainly driven by technology.

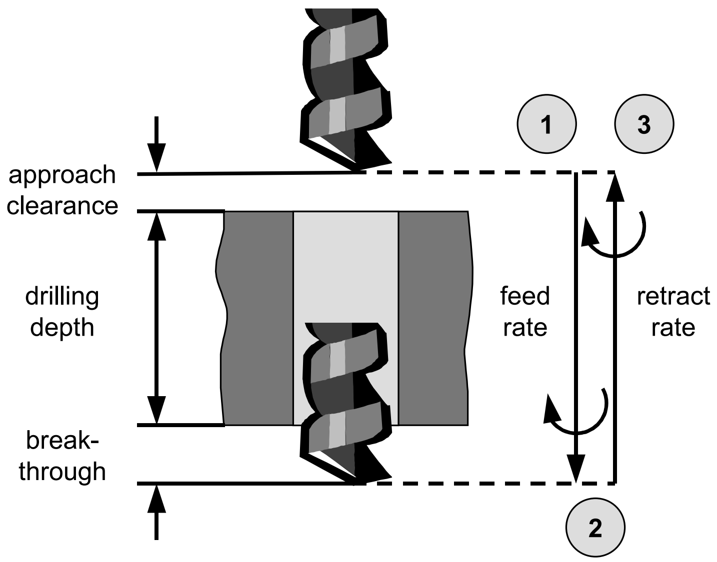

Figure 5.8: Drilling MF.

Depending on the configuration of the machine, milling operations can be performed in 2.5 to 5 degrees of freedom (axes). Pocketing, facing, and contouring operations are prismatic machining operations, which are mostly 2.5-axis machining (i.e. only linear motion possible in the direction perpendicular to the main motion pattern). Then there is 3-axis surface machining, which is dedicated to the machining of 3D parts applying 3-axis machining techniques. Naturally, these multi-axis (e.g. 5-axis) surface machining techniques necessitates the deployment of multi-axis machine tools. One of the most common machining operations for metal parts is pocket milling: removing all the material inside some (arbitrary) closed boundary on a flat bottom of a workpiece to a fixed depth. To completely describe the pocket milling MF, first of all a differentiation is made between roughing, reworking, and finishing and between different machining strategies: outward and inward helical, back and forth, and one way. Furthermore, the MF again is specified by both geometric parameters, e.g. length, width, depth of pocket, depth of cut (geometric parameter mainly driven by technology), and technological parameters, e.g. axial and radial machining feed rates, step-over ratio.

In general, a CAx process model is a sequence of unambiguous instructions to be followed in order to manufacture a part. It is a collection of structured operations which describe the tasks to be performed and the resources needed to do so. The term *structured* means that operations in one setup are grouped into sub-jobs, the sub-jobs of the same workstation are grouped to jobs, and the jobs of one manufacturing task – a logical unit of operations – shape a string of jobs. Sub-job, job, and string of jobs are subsequently termed structuring elements (SE).

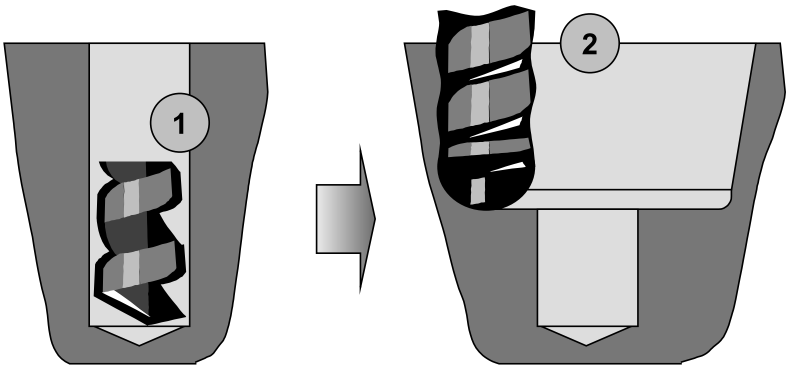

Figure 5.9: Sequence of Operations.

In this thesis, the structuring and sequencing of the operations in the process model are driven by the requirements of cost estimation. In the real world, operations are carried out in a distinct sequence that might be different from the sequence in the model applied here. The example given in figure 5.9 is that of the blind hole and pocket. Here the sequence of operations must not be neglected as it affects the production costs. The feed path for drilling is contingent on the sequence in which the manufacturing operations are carried out: drilling with subsequent pocket milling or vice versa. Furthermore, for the former, the axial approach movement for pocket milling with the machining feed rate may become obsolete. Thus, it is essential that the sequence of operations be considered and a distinction has to be made between predecessor and successor operations, but not necessarily between a direct predecessor or successor.

Apart from the sequence of operations (predecessor relation) there are other aspects which are of importance: for example, concurrency. Concurrency depends largely on the equipment utilized. Operation *b* and operation *c* in figure 5.10 are carried out concurrently.

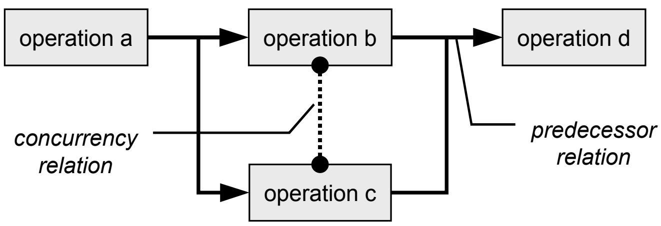

Figure 5.10: Concurrency of Operations.

In conclusion, a rigid adherence to the sequence of operations can be neglected if the following holds true:

- Operations necessary to manufacture the part are applied.
- No superfluous operation is applied.
- Execution time of each operation is unaffected.
- Contemporariness of synchronous operations is preserved.

5.3.2 Resource Consumption

On the one hand, the information that specifies an operation is applied to quantify resource consumption and assign monetary values. On the other hand it is used to assess the utility of an operation of an intermediate most similar case in the retrieval step.

Thus, an operation may consist of six elements:

- Geometric parameters, e.g. drilling depth, nominal diameter of milling cutter, approach clearance.
- Technological parameters, e.g. feed rate.
- Reference to the resources applied, e.g. machine tool, machining tool, supporting supply, auxiliary supply.
- Resource consumption.
- Cost information.
- Organizational parameters, e.g. time stamp, user stamp.

At the least, an operation has a reference to one resource and consequently one resource consumption item and one cost information item. Resource consumption is the time a piece of equipment is deployed or the number of pieces, the volume, the mass, etc. of supporting or auxiliary supply utilized.

Resource consumption is usually computed based on the geometric and technological parameter values of the operation, some of which – especially for machining – correspond to geometric parameters of the finished or the rough parts or the resource, see figure 5.8. In the simplest case, resource consumption is an invariable. The costs that result from the consumption of every single resource are determined separately and are subsequently aggregated.

Since resource consumption of an operation sometimes accrues not only for one part but for several, it has to be divided over the number of parts it affects. Examples for such operations are provided below for illustration:

- Setting-up the machine tool takes the batch size into account.
- Heat treatment considers the number of pieces in an operation.
- A screening test inspects only one out of ten parts.

Thus, a distinction is made between the resource consumption of an operation for the number of parts affected and proportional resource consumption of an operation for one single part. The functions to calculate the resource consumption in total or proportionally are proper to the operation. The number of pieces in an operation may be again a function of the geometric parameters of the part.

Synchronous operations consume equipment as well as operating and auxiliary supply. Care must be taken that multiple charging for resource consumption does not occur. This is the case, in particular for synchronous operations. The procedure in this case would be to assign a monetary value to the resource consumption applicable to the longer operation time.

5.3.3 Process Reference Model

The process reference model is the abstract model that describes the structure of the process class model. It incorporates the objects of the CAx process model and their relations. The key components of the process reference model are the abstract classes operation and structuring elements, which both can be further specialized. A process consists of multiple operations. An operation itself has predecessor and concurrency relations with other operations and structuring relations with structuring elements. The SEs themselves have predecessor relations, concurrency relations, and structuring relations. From all this, the process class model for a certain domain is constructed. Figure 5.11 sets out the process reference model.

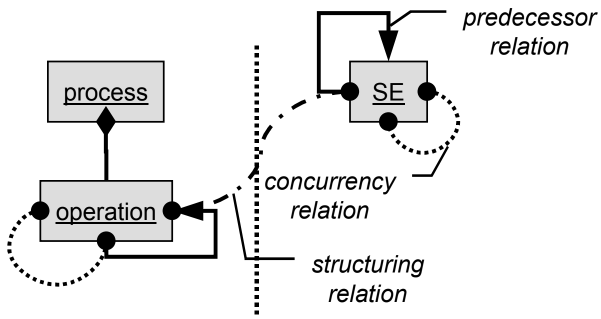

Figure 5.11: Process Reference Model.

5.4 Resource Model

In the scope of this thesis, a resource is a physical entity that is required to execute a certain operation.

As a rule, three types of resources are distinguished:

- Equipment.
- Operating supply.
- Auxiliary supply.

Operating and auxiliary supply support the execution of a manufacturing task. An element of equipment is, for example, a machine tool, a tool, a jig, or a measuring instrument (DIN 199-2–1977). Operating supplies are necessary to produce the product but are not contained in it afterwards, e.g. soldering paste, cleaning material (DIN 199-2–1977). For the production and the intended use of the part, auxiliary supplies fulfill a predefined function, afterwards becoming part of the final product, e.g. coating material (DIN 199-2–1977). It is assumed that an operator with a certain qualification is allocated to a working place.

For cost estimation, only the subset of physical resources is considered:

- The resource utilization of which can be expressed in a monetarily quantifiable resource consumption, e.g. machine tool with a machine hourly rate,
- The application of which drives the operating time, e.g. the diameter of a milling cutter that affects the distance to cover by the tip of the cutter.
- The usage of which is directly or indirectly related to the amount of operating or auxiliary supply and the amount utilized is monetarily quantifiable.

Monetarily quantifiable means that the resource has been assigned a specific cost rate. This in turn means that the resources considered in this thesis directly or indirectly affect the costs of an operation or of the part. Correspondingly, a resource is described by parameters that refer to one of the three categories mentioned above and, in addition, by parameters which serve to check whether a resource is applicable for a certain operation. Other resource parameters are not taken into consideration.

Here, a resource comprises two different kinds of information, whereby not every resource necessarily has both of them. The pertinent information is as follows:

- The specific cost rate.
- The capability of the resource.

A specific cost rate is applied to allocate a monetary value to the resource consumption. The description of the capability of a resource is utilized either to compute the resource consumption (operating time, amount of supporting and auxiliary supply consumed) or to check its applicability for a certain operation. The capabilities of a resource are of either geometric or technological nature. Hence, resource information comprises economic as well as engineering aspects.

The costs as a result of the monetary quantification of the resource consumption may be given as follows:

$$costs = basic\ value \cdot specific\ cost\ rate \tag{3}$$

The basic value, i.e. the resource consumption, has to be estimated as set out in the previous section. The specific cost rate is assumed to be given. For direct manufacturing costs, the vast amount of money usually accrues through usage of equipment (machine tool, operator, tool etc.) rather than usage of operating and auxiliary supply, which are generally part of the overhead rates. In any case, the cost estimation model should be able to take operating and auxiliary supply that have a specific cost rate into account.

For cost estimation it is crucial that the resource model is available – otherwise cost estimation is simply not possible. While this usually holds true, in volume production this is sometimes more difficult, because the resources are developed in parallel or with a slight delay to the product development process. A possible way out is to utilize current resources or to estimate the capabilities and the cost rates of these future resources. Of course, in that case the accuracy declines.

Since only technological planning is carried out, the capacity of the resource is not considered.

The resource reference model depicted in figure 5.12 consists exclusively of the abstract class resource, which has relations of type *resource relation*. A specialization of the resource relation is the *necessary relation* which describes that, in order to able to properly apply a resource, a further resource has to be applied. Again, this resource has a specific cost rate.

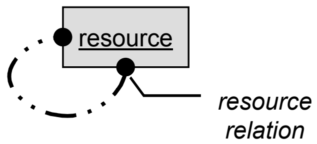

Figure 5.12: Resource Reference Model.

5.5 Representation of Generalized Manufacturing Knowledge

5.5.1 Generalized Manufacturing Knowledge

The cost estimation model developed in this thesis relies on both specific and generalized knowledge. The intention of this chapter is to describe the procedure whereby explicit domain knowledge, i.e. generalized manufacturing knowledge, can be represented in the integrated PPR class model.

For generalized knowledge, a distinction is made between two types:

- Manufacturing knowledge.
- Manufacturing planning knowledge.

Manufacturing knowledge is company-specific and not inherent to the cost estimation model in advance. It comprises knowledge about the specific domain, i.e. manufacturing engineering, and is contained in the knowledge base. This knowledge has to be maintained to keep up with the changes and extensions of the domain. In contrast, manufacturing planning knowledge is compiled knowledge and not company specific. It is the procedural knowledge applied to guide the generation of the process instance model. Manufacturing planning knowledge carries out activities such as grouping of operations and unifying machine tools.

As stated, integration relations in general combine abstract and instantiable classes of the three partial models: product, process and resource. Generative IRs, in particular, comprise the knowledge for automatically instantiating the related objects and their relations. Informational IRs, in contrast, describe only logical relationships. Moreover, to further specify generative IRs, the following additional relation types are introduced:

- Relation-to-part relation.
- Exclusionary relation.

- Concurrency relation.
- Structuring relation.

The domain expert has to define these relations in the integrated PPR class model, which is more comprehensible than representing this knowledge completely in generative IRs. The relation types will be further detailed in the subsequent chapters.

The relations in the integrated PPR class model shape manufacturing knowledge. In contrast to this, manufacturing planning knowledge is implemented in the system in advance.

5.5.2 Generative Integration Relation

Generative integration relations encompass the knowledge for automatically instantiating objects and are able to relate classes:

- A part class and an operation class.
- A form DF class (or GE class) and an operation class.
- An operation class and a resource class.
- A resource class and an operation class.

Generative IRs are directed relations. Figure 5.13 shows them on the class level as described above and how they are instantiated into the integrated PPR instance model. For reason of clearness, solely one instantiation relation is depicted.

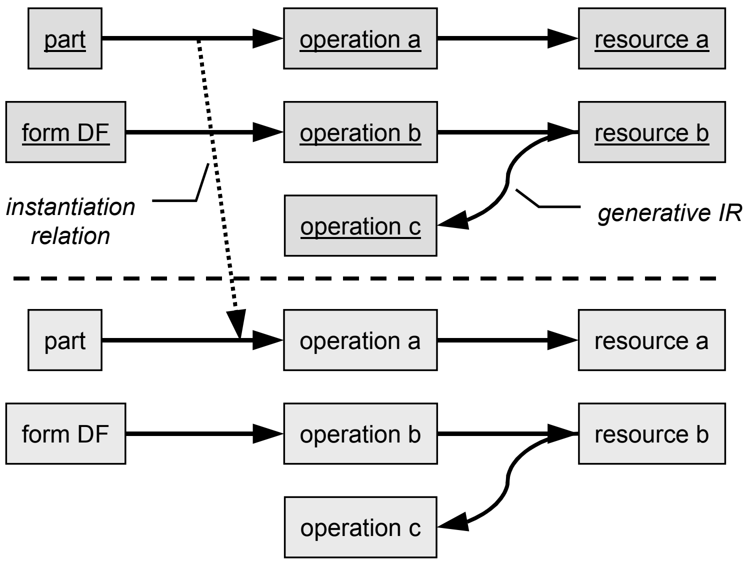

Figure 5.13: Generative IRs.

To make the generative IRs in the figure above more clear: The part, a turbine disc, for example, necessitates the operation *marking*. This operation *a* is carried out on the working place resource *a*. The form DF, an element relevant to the part's security, is processed by the operation *b*, segregated zone etching, which is carried out on resource *b*, which again necessitates the operation *c*, setting up the etching machine.

The relations are of type *m:n*. Thus, *m* objects can be related to *n* objects using generative IRs. Figure 5.14 sets out an example. Two different types of flanges – form DF *a* and form DF *b* – both necessitate the two operations steel shot blasting (operation *a*) and mounting and removing the protective device (operation *b*).

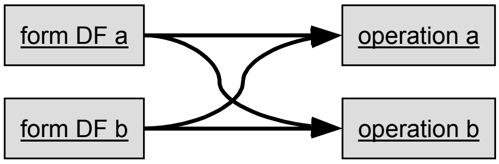

Figure 5.14: m:n Relations.

Generative IRs can start from an abstract class but must end at an instantiable class as depicted in figure 5.15. Thus, the generative IR does not need to contain the information as to which instantiable class has to be specified out of the abstract class and has to be finally instantiated. This allows the comprehensiveness of the knowledge in a generative IR to be limited – which makes it more user-friendly.

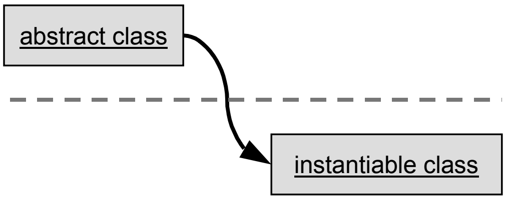

Figure 5.15: Generative IR Between Abstract and Instantiable Classes.

Informational and generative IRs are mutually exclusive. For this reason it is not permissible to define a generative and an informational IR starting from one object (see figure 5.16). This prescription prevents operations from being instantiated on the basis of generalized manufacturing knowledge and being reused from the case base. The case where operation *a* (e.g. drilling) and *b* (e.g. circular milling) do not exclude informational and generative IRs would lead to redundancy and thus to overly high costs in the cost estimate. Two possible workarounds – not further pursued in this work – would allow the coexistency solely for dissimilar operations *a* (e.g. deburring) and *b* (e.g. drilling) or to prioritize one type of relation.

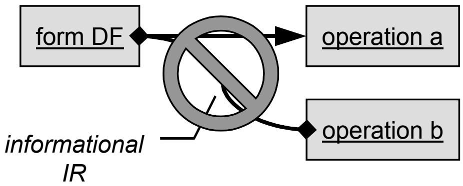

Figure 5.16: Exclusion of Informational and Generative IRs.

In contrast, a generative IR that starts from an object at which an informational IR ends may be defined. However, this is solely the case for a generative IR between a resource and an operation. In all other cases a generative IR may not start from the object where an informational ends. Again, operation *b* is, for example, setting up the milling machine.

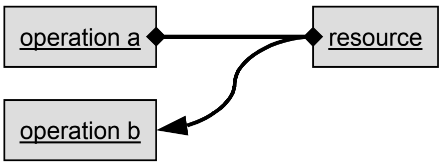

Figure 5.17: Object with Generative and Informational IRs.

The knowledge concerning the instantiation of generative IRs is represented using if-then rules or constraints. The next figure gives an example.

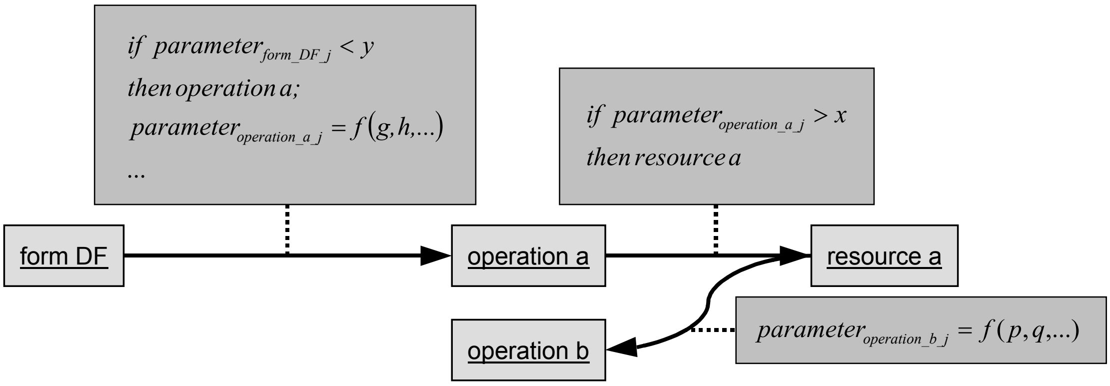

Figure 5.18: Knowledge Representation in Generative IRs.

5.5.3 Relation-to-Part Relation

Form DFs and GEs are both constituents of parts. Yet, the same DF class or GE class can be instantiated in different types of parts. Therefore, a distinction has to be made for which part types each of the generative IRs starting from this DF class holds true. Hence, the generative IRs described above need to be further refined with respect to the type of part for which they hold true: and a relation-to-part relation is applied. A relation-to-part relation specifies that, depending on the type of part, only a limited number of generative IRs can be instantiated.

Relation-to-part relations specify generative IRs that relate to the following:

- A form DF class (or GE class) and an operation class.
- An operation class and a resource class.
- A resource class and an operation class.

Relations that start from a less abstract class are given higher priority compared to relations starting from more abstract classes. Figure 5.19 gives an example. Two alternative resources – resource *a* and resource *b* – are shown for operation *a*. The resource to be chosen depends on the type of part. If the part is of type part *a*, resource *b* and operation *b* would be applied. In contrast, part type *b* would instantiate resource *a*.

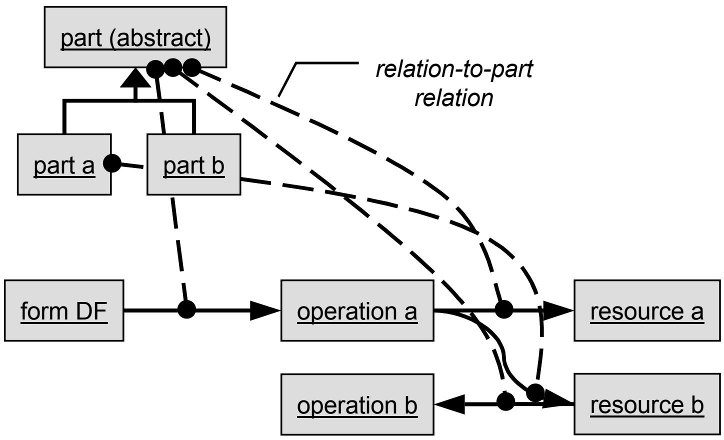

Figure 5.19: Relation-to-part Relation.

5.5.4 Exclusionary Relation

Comparable operations (e.g. vibratory grinding and polishing) allocated to the part and also to its constituents obtain redundant results. Thus, an exclusionary relation has to be introduced. An exclusionary relation defines that only one out of two IRs can be instantiated. It is applied both to avoid redundancy in the process class model and to include a favored operation and dismiss another.

Exclusionary relations can only be defined between IRs that relate objects of the product and the process partial model. If this holds true, the exclusionary relations can be defined as follows:

- Between two generative IRs.
- Between a generative and an informational IR.

Exclusionary relations for generative IRs can only be defined between generative IRs which start from different objects, at least one of which has to be a part class, either abstract or instantiable (see figure 5.20). They need not necessarily end at different objects.

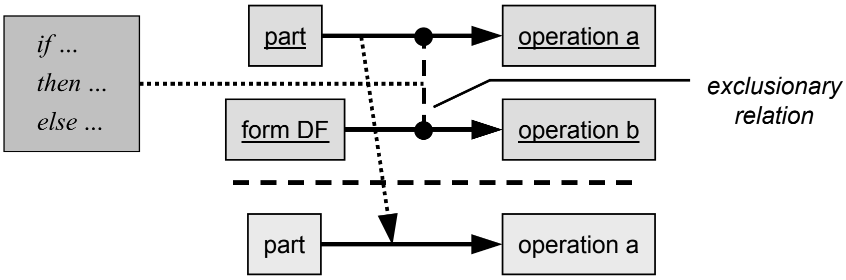

Figure 5.20: Exclusionary Relations Between Generative IRs.

Exclusionary relations for generative and informational IRs can only be defined between IRs which start from different objects, at least one of which has to be a part class, either abstract or instantiable (see figure 5.21). In contrast to exclusionary relations between generative IRs, they need to end at different objects.

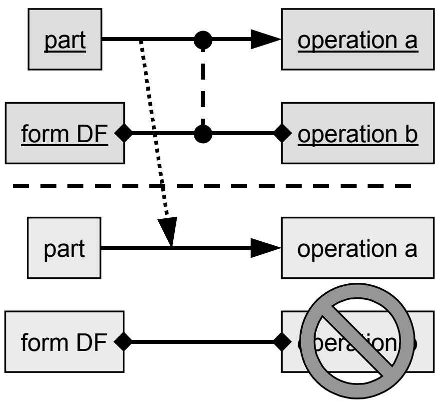

Figure 5.21: Exclusionary Relation Between Generative and Informational IRs.

5.5.5 Concurrency Relation

Operations which are carried out synchronously and which apply the same resource accrue lower costs than they would if carried out sequentially. (In this case the resource consumption is of type *time*) Thus, a concurrency relation has to be introduced. A concurrency relation prescribes that two operations be carried out synchronously and not sequentially.

A concurrency relation has already been introduced in chapter 5.3.1 and solely applied to indicate concurreny of operations on the instance level. This chapter now applies concurrency relations on the class level to represent manufacturing knowledge. Nevertheless, the result after instantiating the relation, either manually or automatically, is the same.

Concurrency relations to further specify generative IRs can only be defined as follows:

- Between two generative IRs which start from the same part class and end at two different operation classes.
- Between two relation-to-part relations, each of which specifies a generative IR that combines a form DF class and an operation class (see figure 5.22).
- Between a relation-to-part relation that specifies a generative IR which combines a part class and an operation class and a generative IR which combines a form DF class and an operation class.
- Between two structuring elements of the same type.

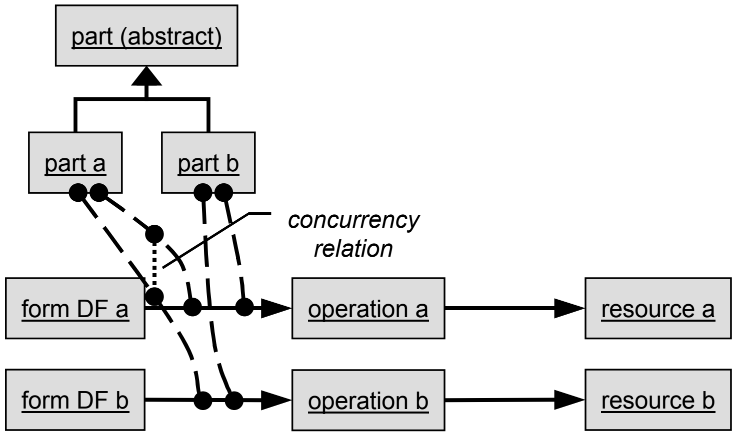

Figure 5.22: Concurrency Relation.

If, in figure 5.22, the part is of type *a* and has form DFs of type *a* and *b*, operations *a* and *b* would be carried out in parallel, whereas for part type *b*, the operations would be carried out sequentially.

5.5.6 Structuring Relation

For cost estimation, structuring elements are applied in the integrated PPR class model to group operations, for example, in order to define concurrency. A structuring relation defines the relationship between generative IRs, relation-to-part relations or between structuring elements. The structuring relations have in common that they end at a structuring element. In more detail:

- A structuring relation that starts from a generative IR, which starts from the part class and ends at an operation class.
- A structuring relation that starts from a relation-to-part relation, which starts from a part class and ends at a generative IR.
- A structuring relation that starts from a structuring elements and ends at a (different) structuring element.

Figure 5.23 gives an example.

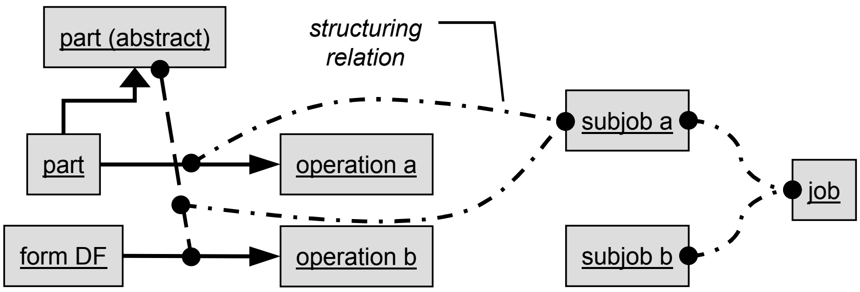

Figure 5.23: Structuring Relations.

5.6 (Re)Use of Specific Manufacturing Knowledge

For the (re)use of specific manufacturing knowledge two types which have to be represented in the knowledge base are distinguished:

- The specific manufacturing knowledge itself.
- The knowledge about manufacturing similarity.

Knowledge about manufacturing similarity is subsequently synonymously termed the *similarity measure*.

Specific manufacturing knowledge is either episodic or prototypical. Episodic knowledge is generated in everyday work in operative, integrated CAx systems. It is represented in the PPR model and makes up the data resource. In contrast, prototypical knowledge is explicitly defined by the user of the case-based application. Episodic knowledge is reused whereas prototypical knowledge is termed to be used.

Knowledge modeling – either initial modeling or knowledge maintenance – for the case-based application is an approach of knowledge acquisition (mainly manual) and machine learning (mainly automatic). The modeling of episodic knowledge is of type *machine learning*, the modeling of prototypical knowledge is of type *knowledge acquisition* and the modeling of the similarity measure is either knowledge acquisition or machine learning.

5.6.1 Retrieve Step

The retrieve step is concerned with searching the case base to find the most adequate operations and resources to manufacture the part. The objective of the similarity assessment as part of the retrieve step is to find the *n* most similar cases in the case base. The quality of each solution is subsequently evaluated using organizational parameters. In the end, the case that is most adequate to solve the current problem is chosen.

Similarity assessment has to take into consideration that manufacturing a form DF not only depends on its characteristics and on its technological DFs but also on adjacent form DFs including their technological DFs. For example, the form DF through hole is child to the form DF pocket. The form DF pocket again has the rough part form DF pre-cast pocket as depicted in figure 5.24. These form DFs, together with their technological DFs, construct so-called *compounds*. For every compound in the CAD product model, the most similar compound in the case base has to be retrieved. In the example given, the approach direction, i.e. orientation of the form DFs – the pocket DF and the through hole DF – is the same. Similarity assessment considers the approach direction of the form DFs but not their position and accessibility. Thus, a compound of a pocket DF and a through hole DF which have opposed approach directions is different from the compound in figure 5.24.

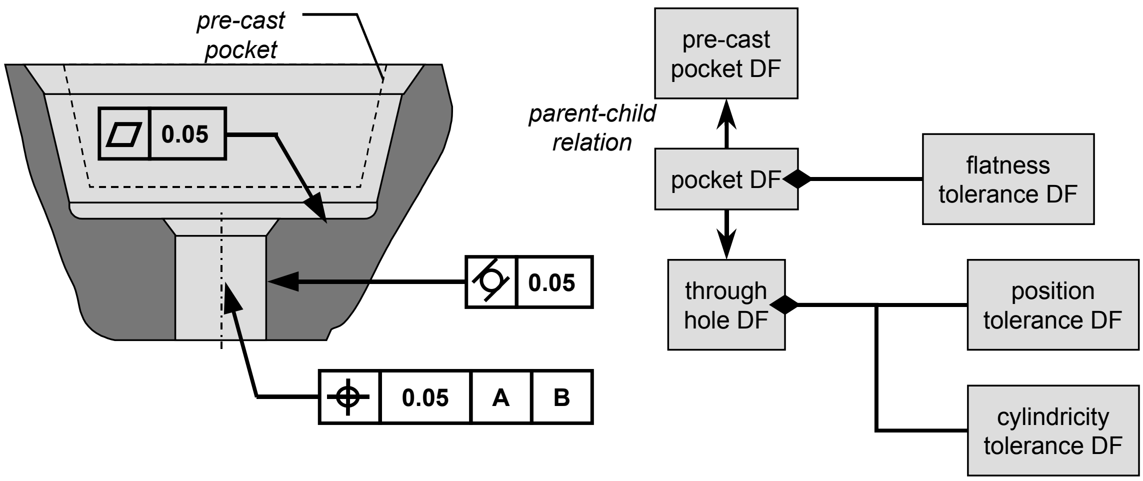

Figure 5.24: Compound in the CAD Product Model.

The objects that constitute the compound are not equally important and therefore have to be weighted. This thesis proposes to weight only the form DFs and its technological DFs but not the form DFs against each other in a compound. Again referring to the above example, this means that the similarity between the compound of pocket DF and through hole DF in the CAD product model and compounds of pocket DF and through hole DF in the case base is assessed. In detail, the similarity between the CAD through hole DF and the through hole DFs in the case base, the similarity between the CAD cylindricity tolerance DF and the cylindricity tolerance DFs in the case base, and the similarity between the CAD position tolerance DF and the position tolerance DFs in the case base are evaluated. The same is done for the pocket DF and its technological DFs. The results of the similarity assessment for the form DFs and their constituting technological DFs are weighted. This is done by computing the similarity between a CAD form DF and a form DF in the case base using the weighted sums of the single similarities. Of course, only the subset of the cylindricity and position tolerance DFs that suffice to be from the selected compound and refer to the same GE is applied. Similarity between two objects *a* and *a**, e.g. form DFs, technological DFs, is computed as set out in chapter 5.7. Figure 5.25 gives an overview: for reasons of clarity, the similarity for the flatness and the position tolerance DFs and the through hole DF are left out.

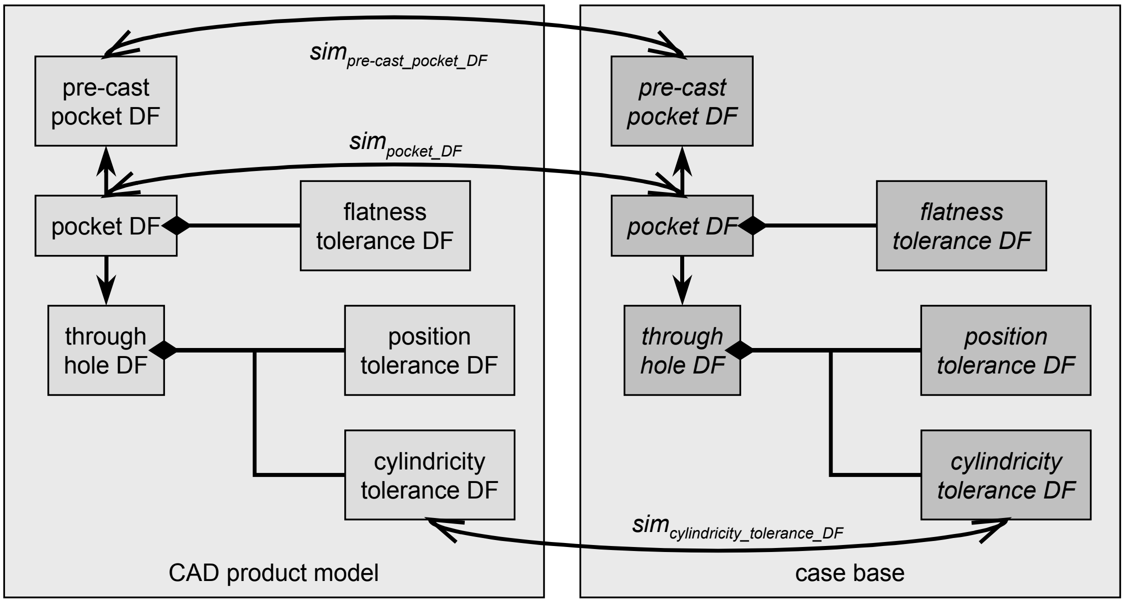

Figure 5.25: Object Similarities in a Compound.

If the one form DF has a technological DF, e.g. a cylindricity tolerance DF, and the other form DF does not, yet the similarity between the two objects is nevertheless to be assessed, the general tolerance of the form DF without a technological DF is applied. Depending on the parameters of the form DF, a virtual technological DF is generated. Thus, the similarity between the two technological DFs can be evaluated and applied to assess the similarity between the two form DFs.

The similarity of the compound is the unweighted sum of the similarities of the pocket DF and the through hole DF. Figure 5.26 shows that, for the example pocket and through hole, the position tolerance DF and the through hole DF have a weighting coefficient *g* unequal to 1, while the pocket DF and the through hole DF do not (*g=1*).

The benefit of starting with a compound is that this procedure takes into consideration that not every object is manufactured independently from any other object. Yet, at the same time, it involves risk because it limits the number of available cases in the case base. Therefore it may not be the most adequate solution. Hence the retrieval strategy has to begin assessing the similarity for a compound while in parallel ignoring the compound and assessing the similarity of form DFs in the CAD product model with form DFs of the same type in the case base. Of course, the result of the latter is usually better since a greater number of cases is available for the similarity assessment. Therefore, a reduction factor has to be applied to make the result comparable to the result derived out of the compound.

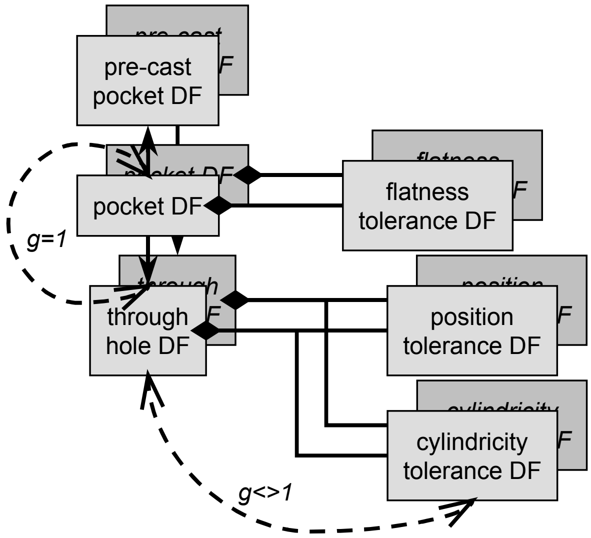

Figure 5.26: Weighting Coefficients for Object Similarities in a Compound.

The similarity assessment starts with evaluating the similarity of the overall compound. If there is no compound in the case base that has at least one operation which is influenced by at least one geometric parameter of every *m* form DFs of the compound, similarity assessment has to proceed with the similarity assessment of a compound that consists of *m-1* objects (parent-child relation) or with regarding every form DF with a parent-child relation in the compound independent of any other form DF. If the case retrieved has at least one operation that is influenced by at least one geometric parameter of every remaining form DF, proceeding with a compound of *m-2* objects is skipped. Otherwise the same procedure is run through again.

Below, three examples are given: in parallel to the similarity assessment of every compound, similarity assessment is carried out for each pocket independent of the interaction.

The first example in figure 5.27 shows three pockets – pocket *a*, pocket *b*, and pocket *c* – that are nested in each other. This means that the pockets are arranged in a chain-like structure. The similarity assessment process starts to figure out any compound of three pockets – pocket *a**, pocket *b**, and pocket *c** – in the case base that has at least one operation with at least one geometric parameter of every pocket. If there is one operation that fulfills the requirement, the similarity $sim_{a\text{-}b\text{-}c,\ a^*\text{-}b^*\text{-}c^*}$ is computed. If not, the similarity assessment process goes on and tries to figure out any compound of two pockets *a* and *b* that fulfills the requirement that the case has at least one operation with at least one geometric parameter of pocket *a** and one of pocket *b**. If there is a case that fits these requirements, the similarity $sim_{a\text{-}b,\ a^*\text{-}b^*}$ is computed. If not, the similarity sim_{a,p^*} between pocket *a* and pockets *p** in the case base is computed, as well as that between the similarities sim_{b,p^*} and sim_{c,p^*}. In order to limit the complexity of the retrieve step, no check is made whether there is a compound of pocket b* and c* in the case base that has at least one operation with at least one geometric parameter of every pocket.

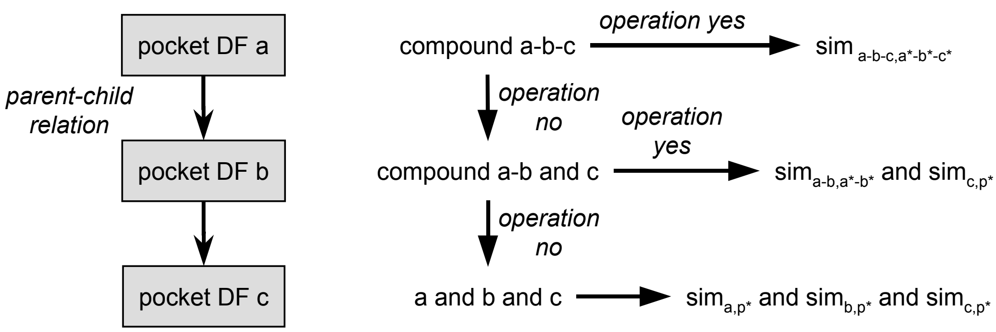

Figure 5.27: Three Pockets Nested in Each Other.

Figure 5.28 shows two rectangular pockets – pocket *b* and pocket *c* – that are nested in a third pocket, *a*. If there is no compound that fulfills the requirement of at least one operation as described above, similarity assessment for this tree-like structure proceeds with trying to figure out a compound that fulfills this requirement for a branch of the tree. Thus, for every branch, the procedure set out above is again applied: the similarity is assessed between a compound of pocket *a* and *b* and compounds of pockets *a** and pockets *b** in the case base, and the same is done between the compound of a and c and compounds of a* and c* in the case base. If the requirement is fulfilled, the similarity $sim_{a\text{-}b,a^*\text{-}b^*}$ or $sim_{a\text{-}c,\ a^*\text{-}c^*}$ is computed. If, for both branches *a-b* and *a-c*, a compound is found, similarity is computed regardless. Reuse has to take into account that the operation to manufacture the parent pocket is taken from the branch, i.e. either *a*-b** or *a*-c**, that has the higher similarity measure.

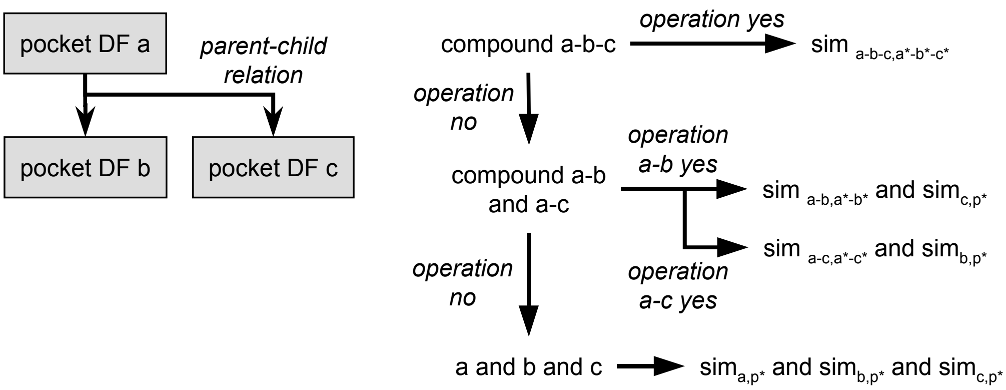

Figure 5.28: Two Pockets Nested in a Third Pocket.

Figure 5.29 depicts two intersecting pockets *a* and *b*. Similarity assessment starts with assessing the similarity between the compound *a-b* and compounds *a*-b**. If no operation that fulfills the operation requirement is retrieved, similarity is assessed between pocket *a* and pockets *p** and between pocket *b* and pockets *p**, which is the same as assessing every pocket independently from any other. Similarity assessment has to take into account the angle of intersection.

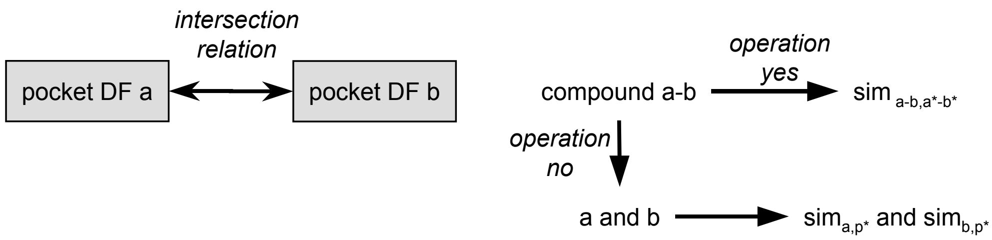

Figure 5.29: Two Intersecting Pockets

The procedure described above presupposes that if there is no interaction relation between the form DFs there is also no operation for a form DF that has geometric parameters of another form DF as input. However, this is a simplification as the two holes with the same orientation but on different surfaces do not possess an interaction relation. Nevertheless, they may be manufactured in a single operation applying a multi-spindle drilling machine.

The total number of parts to be manufactured and the batch size usually drive manufacturing – and especially the manufacturing methods applied. This thesis assumes that, first, product development is aware of what affect they have on the product specification and, second, that the total number of parts to be manufactured and the batch size of the parts in the case base are comparable. Thus, the retrieval process does not consider these factors.

The quality of the proposed solution in the most similar cases has to be evaluated. Therefore, the organizational parameters of an operation instance are applied. These are, for example, the following:

- Time stamp that contains the point of time of instantiation or modification.

- User group identification.

The time stamp is applied to evaluate the relevance of the operation, while the identification of the user group, e.g. novice, expert, is applied to evaluate the reliability. However, discriminating the users may not be possible in a productive application. The organizational parameters are applied to generate a measure similar to the similarity measure, the value range of which lies between 0 and 1. Time stamps that date far back in time or an inexperienced user group move the measure toward 0 whereas time stamps that refer to the near past or an experienced user group increase the measure towards 1. The result of the similarity and of the evaluation process have to be adequately combined. Therefore, again a weighting coefficient is introduced.

5.6.2 Reuse Step

For reuse, a distinction is made between operations for which resource consumption is independent from the geometric parameters of the part and resource consumption that is a function of geometric part parameters. For the former, reuse is to take the retrieved solution as the solution to the current problem, i.e. solely copying the solution from the case base. For the latter, reuse necessitates that the retrieved solution be modified, i.e. by adapting it to the current problem without composing the solution of multiple cases, in order to compute resource consumption based on up-to-date information. To do so, the geometric parameters of an operation that correspond to the geometric parameters of form DFs are replaced by the parameters of the current CAD product model. For both, the number of parts which the operation affects has to either be taken from the current MF, be computed based on current geometric parameters or be derived from the CAD product model. Two examples for reuse by adaptation are given below.

Figure 5.30 depicts the form DF through hole of the CAD product model and the most similar case in the case base. Three operations, drilling, chamfering and reaming (chamfering and reaming are not displayed in the figure below), have been applied to manufacture this most similar through hole. To generate the solution for the through hole in the CAD product model, the three operations are adapted. Thus, for drilling, the technological parameters (feed rate and retract rate), the geometric parameters that do not correspond to geometric parameters of the form DF (approach and breakthrough), and the resources are taken from the most similar case. The geometric parameters (drilling depth) of the drilling operation that correspond to the geometric parameters of the form DF are taken from the through hole DF of the CAD product model. The same is repeated for the chamfering MF and the reaming MF. Hence, resource consumption is computed based on the parameters of the through hole DF of the CAD product model and on the parameters of the MFs of the most similar case in the case base.

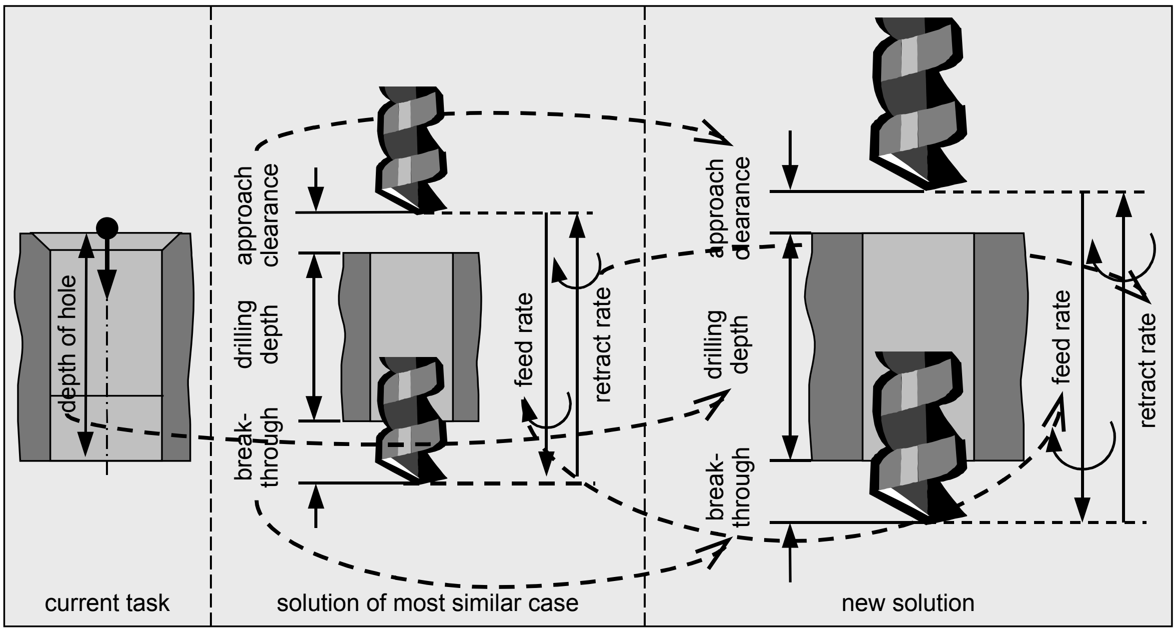

Figure 5.30: Reuse Step for a Single Form DF.

For a compound of for example a rectangular pocket DF and a through hole DF that is nested in the former the operations of the most similar case in the case base are drilling and chamfering to manufacture the through hole and rough and finish pocketing to machine the pocket. The sequence of operations for this solution encompasses drilling, rough pocketing, finish pocketing, and chamfering. Even though each operation is applied to manufacture solely one form DF, for drilling, the geometric parameter depth of the rectangular pocket in the CAD product model is applied for reuse in addition to the geometric parameters of the through hole. The technological parameters are again taken from the MFs.

Applying the resource's capability parameters, the result of the reuse is revised. In case of failure, the solution of the second most similar case is reused and again revised. The retain step is of no relevance here.

5.7 Manufacturing Similarity

5.7.1 Requirements for Similarity Definition

Chapter 3.4 has described the characteristics of similarity. As shown, similarity is always relative to some aspects so that, additional, domain-specific knowledge is required to distinguish whether two different attribute values mean the same, something similar or something completely different. For this reason, similarity has to be defined and represented before it can be applied to assess the similarity between two objects.

There are four key requirements to be met for a manual definition of similarity carried out by the human domain expert:

- Intuitive definition.
- Exact reflection of the manufacturing aspects.
- Easy maintainability.
- Good comprehensibility.

An intuitive definition incorporates that the definition of the manufacturing similarity corresponds to the expert's way of thinking: it must promote the remembering of concrete, historic situations and reflect manufacturing aspects. For manufacturing, a vast quantity of parameters has to be taken into account. Usually, the parameters are not independent of each other, rather they influence each other. So the relationships between the parameters have to be resolved in order to obtain manageable portions for which manufacturing characteristics can be defined. Then, these portions have to be recomposed.

The similarity measure needs to be defined in advance and has to keep track with changes in manufacturing, i.e. once defined it has to be maintained from time to time.

Since the parameters of an object show different attribute types – binary attributes, numerical attributes, which can either be continuous or discrete, and nominal attributes – the definition, representation, and assessment of manufacturing similarity is non-uniform. Thus, for the different types of parameters and whether they show further dependencies, different ways of defining and representing the manufacturing similarity are subsequently presented.

5.7.2 Allocation Function for Numerical Parameters

This chapter addresses how to define, represent, and assess the similarity for continuous, numerical parameters.

The formalization of similarity as a measure that quantifies the degree of similarity between the objects a and a^* is most appropriate. To define the similarity, something that is intuitive and in line with the expert's way of thinking is needed. Thus, with regard to the requirements manifested above, for continuous, numerical parameters an allocation function is applied.

An allocation function represents manufacturing aspects, enabling the quantification of the degree of similarity. Therefore, for an object class of the product partial model, e.g. form DF, technological DF, GE, and a parameter of this object class, and different technological criteria in each case, an allocation function exists. The allocation function assigns to each possible value of the parameter a numerical identification for the kind of use and/or the use or the time and/or quantity used of at least one resource that is applied to manufacture this object class.

The definition of the allocation function starts by manifesting a definition range, i.e. only within this definition range can a variation with respect to the defined criterion be made. Then, for each variation, a parameter value has to be defined. Furthermore, the type and size of variation have to be defined. The allocation function is necessarily monotonic increasing in nature.

Explorative studies show that it is advantageous to provide libraries with ramp and step functions of different type and size as this guides the definition of the allocation function, enhancing its intuitiveness. The definition of the allocation function can guide the following generic question: for which parameter value is there a variation with respect to the selected criterion and what is the type of this variation and its size?

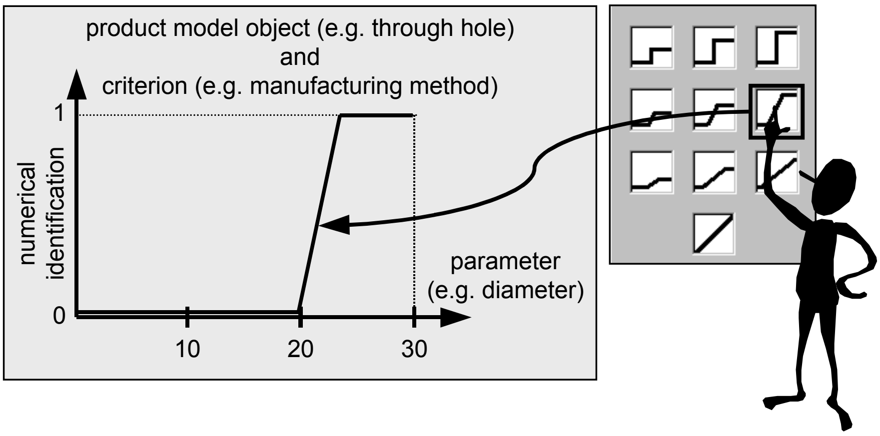

Figure 5.31: Allocation Function.

Figure 5.31 depicts an allocation function in an exemplary fashion. The definition range of the example allocation function is from 0 to 30 mm. As the figure shows, there is a variation for the parameter value 20. If the product model object is a through hole DF, the parameter is the diameter, and the selected criterion is the manufacturing method, then the reason for this variation is that drilling is applied for parameter values smaller than 20, whereas for parameter values between 20 and 24 either drilling or circular milling is applied. For parameter values greater than 24, solely circular milling is applied. A ramp function instead of a step function is applied since there is no abrupt change in the manufacturing method: there is a transition region instead.

Each parameter requires its own allocation function, which can be different for various parameters. The allocation function solely takes into consideration the absolute value of the variation but not the direction. This is done because it is assumed the objects that have a greater distance with regard to their parameter values are less similar than neighboring objects. A possible way out in order to consider the direction is the application of a similarity matrix.

It is assumed that the definition of the allocation for this parameter can be made independently of any other parameter of this through hole. This does not always hold true. For this reason the definition of an allocation function for interacting parameters will be explained in the next chapter.

An allocation function can be defined for different criteria, which can be subdivided into technological (manufacturing method, feed rate, step-over ratio, etc.) and resource-related (machine tool, machining tool, supporting supply, auxiliary supply, etc.) criteria. In conclusion, any of the parameters that specify a process or a resource can be applied as criterion.

The various criteria of a parameter of one object class are not always equally important. Thus, each criterion has its own weighting coefficient. If the criterion c_{ji} is twice as important as criterion $c_{j(i+1)}$ then it must be weighted double; if it is only half as important as c_{ji} then it is weighted by a factor of 0.5. To do so, the allocation function of a parameter is the superposition of the weighted allocation functions of the criteria divided by the sum of the weighting coefficients. The weighting coefficient for a criterion is assumed to be the same within one object. And not only are the criteria weighted, the parameters also possess weighting coefficients indicating their importance.

To assess the similarity between two objects *a* and *a**, the steps set out below have to be followed for every parameter which has an allocation function. Object *a* stands for an object of the current CAD product model, for which the costs are to be estimated; object *a** are objects of the same type in the case base. *A* denotes the object class, which is either instantiated in the CAD product model or stored in the case base.

- The value p_{aj} of the object parameter a_j and the allocation functions defined for A_j are determined.
- For object *a** of the same type in the case base, the value p_{a*j} of the object parameter a^*_j is determined. The allocation function is the same for p_{aj} and p_{a*j}.
- A distance $d_{Aj,cji}$ between the values of these two parameters a_j and a^*_j for criterion c_{ji} is computed, namely by applying the function

  $$d_{A_j,c_{ji}}\left(p_{a_j}, p_{a^*_j}\right) = dist(f_{A_j,c_{ji}}(p_{a_j}) - f_{A_j,c_{ji}}(p_{a^*_j})) \quad (4)$$

  with

  - $d_{Aj,cji}$...distance between the two parameter values p_{aj} and p_{a*j} for criterion c_{ji}.
  - *dist*...distance function.
  - $f_{Aj,cji}$...allocation function for parameter *j* of object *a* or object *a** and criterion c_{ji}.
  - p_{aj} and p_{a*j}...parameter values of the parameters a_j and a^*_j.
- The similarity $sim_{Aj,cji}$ of the two objects *a* and *a** with regard to the parameter *j* and the criterion c_{ji} is determined by the application of the similarity function *sim* to the distance $d_{Aj,cji}$. The result is then given as

  $$sim_{A_j,c_{ji}}\left(d_{A_j,c_{ji}}\left(p_{a_j}, p_{a^*_j}\right)\right) = sim\left(dist\left(f_{A_j,c_{ji}}(p_{a_j}) - f_{A_j,c_{ji}}(p_{a^*_j})\right)\right) \quad (5)$$

These steps are repeated for every criterion of each parameter of this object class. Next, the similarity between the two objects *a* and *a** is computed as the weighted sum of the similarities of the parameters $sim_{Aj,cji}$. The similarity $sim_{a,a*}$ between the two objects *a* and *a** is

$$sim_{a,a^*} = \frac{\sum_{j=1}^{n}\left(\left(\frac{\sum_{i=1}^{m_j}\left(sim_{A_j,c_{ji}} \cdot g_{A_j,c_{ji}}\right)}{\sum_{i=1}^{m_j} g_{A_j,c_{ji}}}\right) \cdot g_{A_j}\right)}{\sum_{j=1}^{n} g_{A_j}} \quad (6)$$

with

- g_{Aj}...weighting coefficient for object class *A* and parameter *j*.
- $g_{Aj,cji}$...weighting coefficient for criterion c_{ji} of parameter *j* of object class *A*.

Taking into account the weighting coefficients for object similarities in a compound the similarity is

$$sim_{a,a^*;b,b^*;...} = \frac{sim_{a,a^*} \cdot g_{a,a^*} + sim_{b,b^*} \cdot g_{b,b^*} + ...}{g_{a,a^*} + g_{b,b^*} + ...} \quad (7)$$

For continuous, numerical parameters it is assumed that symmetry of similarity always holds true. In this thesis, for every allocation function and every similarity measure, the value range is between 0 and 1. Thus, the distance function is

$$dist(x) = |x| \quad (8)$$

and the similarity function *sim* is

$$sim(y) = 1 - y \quad (9)$$

Figure 5.32 shows how the similarity between the two diameter values p_{aDB}=*14mm* and p_{a^*DB}=*22mm* is determined for the form DF through hole with respect to the criterion manufacturing method. In DIN 32869-3–2002 is *DB* the abbreviation for the nominal diameter of a through hole.

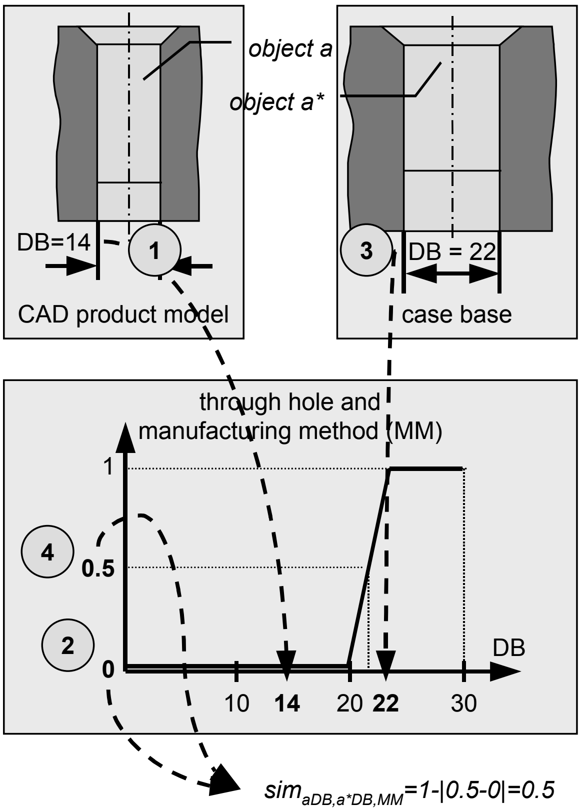

Figure 5.32: Similarity Assessment.

The advantages of such an allocation when used for similarity assessment are manifold:

- It is intuitive to define.
- It does not need an empirical foundation.
- It allows steps to be represented.

In contrast to these benefits, such an allocation function also shows some limitations:

- It is only defined for a limited definition range.
- It is necessarily symmetric.
- Changes are non-directional.

Yet, for the assessment of manufacturing similarity these limitations do not narrow down the comprehensiveness of its application.

As described, a form DF is specified by technological DFs with the DFs impacting on the manufacturing costs differently. For this reason, the form DF and its technological DFs are weighted. In certain cases it is more useful to define the weighting coefficients not for instantiable but for abstract classes, taking into account that allocation functions and weighting coefficients sometimes depend on the type of GE to which they are applied. For example, an allocation function for perpendicularity tolerance DFs for a cylinder is different from the function for a planar face.

5.7.3 Allocation Function for Interacting Numerical Parameters

The previous chapter has described the definition of an allocation function for a single parameter, independent from any interaction with other parameters of the same object. This chapter is concerned with the definition of an allocation function for interacting parameters. The requirements to be met as set out in chapter 5.7.1 still hold true.

Again, the definition of the allocation function starts with the selection of an object parameter, a criterion, and the expression of the definition range. What is different is that, in addition to the object parameter for which the definition range is given, one or more interacting object parameters of the same object are selected and, for each parameter, a discrete parameter value or an interval are chosen. For each variation, a parameter value has to be defined. As presented in chapter 5.7.2, the type and size of variation are used to describe the allocation function. Subsequently, for this specific constellation, but for other criteria, further allocation functions can be defined. Moreover, different discrete parameter values for the interacting object parameters are selected and, again for not necessarily the same criteria, allocation functions can be defined. Weighted allocation functions for the same constellation of parameter values, but for different criteria, are subsequently superimposed to obtain a single allocation function for each constellation.

Figure 5.33 depicts three superimposed allocation functions for two interacting object parameters.

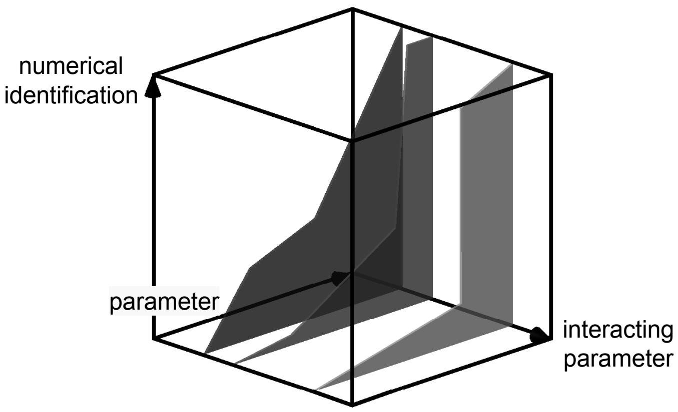

Figure 5.33: Allocation Functions for Two Interacting Parameters.

Again, within the definition range, the allocation function provides a numerical identification for the resource consumption. Yet, this is true only if the parameter value of the interacting parameter, for which solely discrete values are specified, exactly matches. In all other cases the numerical identification has to be derived by interpolation between the allocation functions. Since the function to be derived with regression analysis is not necessarily differentiable and is to provide parameter values between 0 and 1, the simplest interpolation is applied: polygonal sequence. This kind of interpolation allows intervals of interacting parameter values for which the same numerical identification holds true to be described. If applying interpolation other than polygonal sequence, e.g. LaGrange interpolation, (cubic) spline interpolation, this does not hold true. Also, the interpolation of functions with step wise changes in function value would not be possible.

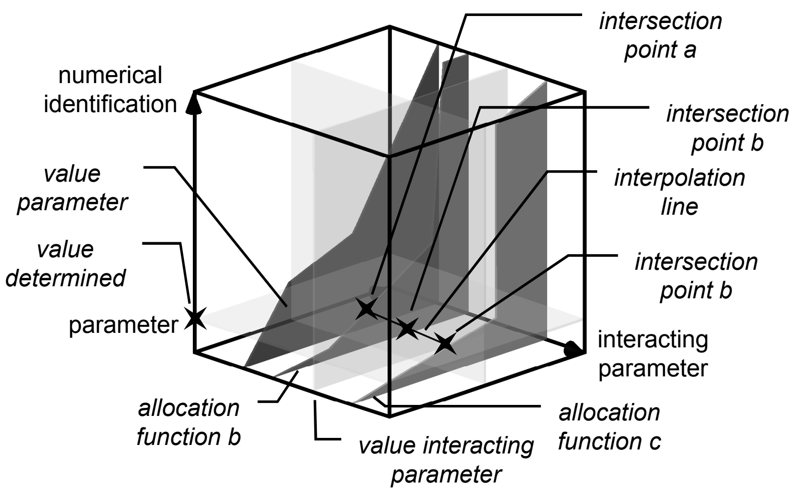

Figure 5.34: Numerical Identification for Two Interacting Parameters.

Figure 5.34 illustrates how the numerical identification is derived for two interacting parameters and three allocation functions. The value of the parameter determines the intersection plane *a,* which generates the intersection point *a* on allocation function *b* and the intersection point *b* on allocation function *c.* The interpolation line combines these two intersection points. The value of the interacting parameter generates intersection point *c,* which provides the numerical identification desired.

Only the determination of the numerical identification differs. The other steps for assessing the similarity between two objects *a* and *a** as presented in chapter 5.7.2 remain the same.

5.7.4 Similarity Matrix for Nominal and Discrete Parameters

To assess the similarity between two objects, a distinction is made between continuous, numerical attributes and other types of attributes. How to derive the similarity measure for the former was explained above. This chapter is concerned with how to define and represent the similarity for nominal and discrete, numerical parameters.

To define the similarity for this kind of parameters, semantic differentials are best suited. Hence, to define the similarity between two parameter values such as between two metrical nominal thread sizes M8 and M10, a discrete number of semantic differentials that reach from completely dissimilar to identical is applied. Since the degree of similarity still needs to be quantified, it is best to formalize the similarity again as a measure. A numerical identification – the similarity measure – is assigned to each semantic differential. The value range is from 0 to 1.

A matrix is used to represent the similarity. The first column contains the object *a* parameters, the first line the object *a** parameters. The object parameters are, of course, identical. The reflexivity, i.e. each object is similar to itself, holds true so that the diagonal line contains the maximum of similarity measure. Furthermore, the symmetry primarily holds true so that, in this case, the domain expert needs to fill in only one half of the matrix. For example, a matrix is used to compare different materials, with deviation as per ISO 286-1–1988, general tolerance as per ISO 2768-1–1989 and ISO 2768-2–1989, surface texture lay (ISO 1302–2002), and – as mentioned – nominal diameters of thread (e.g. DIN 13-1–1999). Figure 5.35 portrays the definition and representation of similarity of materials with regard to machinability in a similarity matrix. Furthermore, the figure below depicts how to assess the similarity between an object *a* with the material parameter value *C45* and an object *a** with the material parameter value *St52*.

product model object (e.g. through hole DF) and
criterion (e.g. machining feedrate)

*object a** / object a	*$p_{a^*j}1$ (C45)*	*$p_{a^*j}2$ (St52)*	*$p_{a^*j}3$ (GGG50)*	...
$p_{aj}1$ (e.g. C45)	similarity measure (identical)	**similarity measure (e.g. dis-similar)**	similarity measure (e.g. dis-similar)	—
$p_{aj}2$ (e.g. St52)	—	identical	similarity measure (e.g. dis-similar)	—
$p_{aj}3$ (e.g. GGG50)	—	—	identical	—
...	—	—	—	identical

Figure 5.35: Similarity Matrix.

As shown in the example above, analogous to the definition and representation of similarity for continuous, numerical parameters using an allocation function, for discrete, numerical and for nominally-scaled parameters, different criteria can be applied for one object parameter to define the similarity with regard to them. Again, each criterion of an object parameter has its own weighting coefficient. The weighted similarity measures for two object parameter values with regard to the criteria are superimposed to one similarity measure for this parameter value. But, in order to enable further maintenance and to support comprehensibility, the single similarity measures for the various criteria are maintained.

For the representation of a similarity measure in a matrix, the steps to derive the similarity between two objects *a* and *a** as described for an allocation function in chapter 5.7.2 can be skipped for a similarity matrix. The similarity measure can be directly determined and is subsequently applied to compute the similarity between the two objects applying equation (6).

Of course, to assess the similarity between two objects *a* and *a** can incorporate the steps to derive the similarity measure for continuous, numerical parameters as set out in chapter 5.7.2, for interacting parameters as cited in chapter 5.7.3, and for discrete numerical and nominal parameters as introduced in this chapter. The single similarity measures again are input for equation (6).

5.7.5 Reducing the Effort for Similarity Definition

The previous chapters have described how similarity is defined, represented, and assessed. Even though the knowledge acquisition bottleneck is not shifted from the acquisition of if-then rules for traditional knowledge-based systems to the definition of the manufacturing similarity for case-based applications, it is advantageous to search for further possibilities to cut the effort for knowledge acquisition and maintenance. One such possibility is to inherit similarity measures. Another possibility is to determine from the data resources a reasonable correlation between product characteristics and manufacturing or manufacturing costs.

Chapter 5.2 has introduced super-class-sub-class relations between objects and has described the inheritance of properties from the super-class to its sub-classes. To advantage, allocation functions, similarity matrices, and weighting coefficients defined for a super-class are inherited to its sub-classes. If the sub-class has its own similarity measure which is different from that of the super-class, the super-class is ignored. This means that, for similarity definition, not only an instantiable but also an abstract class can be chosen. Furthermore, a similarity definition module in a case-based application has to offer the functionality of copying allocation functions, similarity matrices, and weighting coefficients from one object to another. Both inheritance and copying similarity measures facilitate manual manufacturing similarity definition and decrease the effort needed.

In the following, a concept will be shown to determine reasonable correlation in the data sets. The correlation can be used either to guide the user by indicating parameters that have a great impact on resource consumption or to almost entirely remove the burden from the user by automatically scanning the data sets and determining allocation functions, similarity matrices, and weighting coefficients. For parameters that greatly impact resource consumption, it is indispensable to define the manufacturing similarity. Parameters that hardly impact resource consumption can be skipped in favor of a reduced input time.

The next paragraphs will portray how the weighting coefficients can be derived from the data sets. The weighting coefficients can serve to indicate the most important parameters with respect to resource consumption in order to guide the user in the definition as described above. Or they can be accepted as the coefficients to weight similarities either of criteria, of object parameters or of objects in a compound.

The data resources can be regarded as matrices of type v,w (v rows and w columns). A row corresponds to a data set and a column corresponds to a parameter and its parameter values in the data sets.

In general, determining weighting coefficients for numerical parameters (continuous and discrete) necessitates the following steps (see figure 5.36):

- Selection of the appropriate data sets and parameters of the object under consideration.
- Generation of dot clouds in cost-parameter diagrams.

- Computation of a regression line for every dot cloud.
- Determination of the definition range.

The weighting coefficients correspond to the absolute value of the inclination of the regression line.

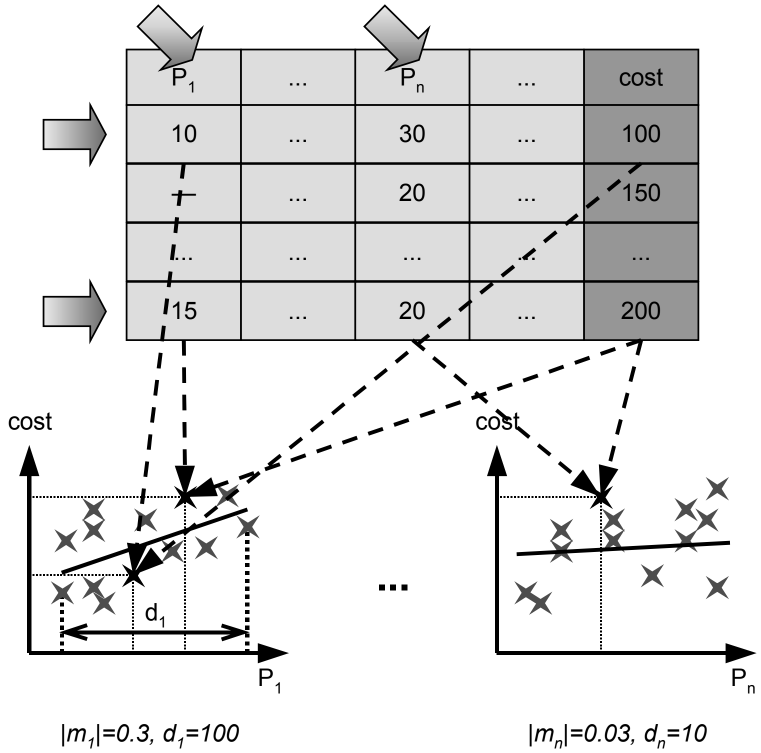

Figure 5.36: General Procedure to Determine a Weighting Coefficient.

The vertical axis in figure 5.36 corresponds to the total costs of an operation and the horizontal axis to the item for which the weighting coefficient is to be determined. The more the inclination of the regression line differs from 0, the greater the impact of the item on the costs. It is irrelevant whether this is a positive or negative inclination. Nor is for the determination of the weighting coefficient the variance for the dot clouds of primary interest. However, it does have to be assured that the inclinations are comparable despite the various units of the parameter values of the items and their value range. This thesis proposes to scale the parameter values on the horizontal axis to an invariable definition range, e.g. 1.

The variance of the dot clouds does not drive the value of the weighting coefficient: it indicates the probability of the correctness of the weighting coefficient. Thus, for each weighting coefficient the variance of the dot cloud is computed:

$$\sigma_j^{\;2} = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - x)^2 \qquad (10)$$

with j indicating the parameter, x_i being the cost value of data set i, and x the cost value computed by applying the regression line and the corresponding parameter value for this data set i and n the total number of selected data sets. The manufacturing engineer can apply the variance when validating or modifying automatically determined weighting coefficients.

For nominally-scaled parameters the determination of the weighting coefficients differs. Figure 5.37 shows the necessary steps for this kind of parameter: make a group for every parameter value on the horizontal axis, generate the point for every data set and write it down in corresponding groups at the correct position on the vertical axis. An arithmetic median is computed for the data points in each group. For two arithmetic medians at a time, a ratio factor r_i is computed until there are no two parameter value groups left that do not possess a ratio factor. A ratio factor is the quotient of the bigger parameter value divided by the smaller parameter value. For the ratio factors, the arithmetic median is again computed.

$$m_{j_{nominal}} = \frac{\sum_{i=1}^{n} r_i}{n} \tag{11}$$

The arithmetic median of the ratio factors has to be scaled in order to fit the inclinations of the numerical parameter values. What a proper scale factor is has to be determined on the basis of real data. The scaled arithmetic median of the ratio factors is again used to determine the weighting coefficients of the items. The variance for nominally-scaled parameters is the arithmetic median of the variances in each group.

$$\sigma_{j_{nominal}}{}^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} \sigma_i{}^2 \tag{12}$$

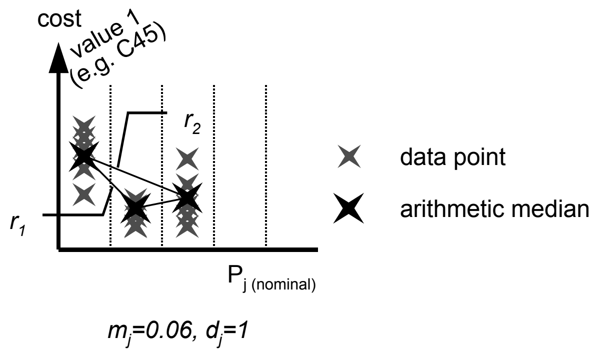

Figure 5.37: Determination of Ratio Factors for Nominal Parameters.

Thus follows for a weighting coefficient g_j:

$$g_j = \frac{\frac{|m_j|}{d_j} \cdot s_j}{\sum_{j=1}^{n} \left(\frac{|m_j|}{d_j} \cdot s_j \right)} \tag{13}$$

with *j* indicating the parameter or the criterion, *m* being the inclination of the regression line for numerical parameters or the arithmetic median of the ratio factors for nominally-scaled parameters, and *d* the definition range for numerical parameters. For nominally-scaled parameters *d* is 1. And *s* is the scale factor to make the value of different units comparable. The weighting coefficient *g* can be determined either to weight the parameters of a product model object, to weight the different criteria of a product model object or to weight the technological DFs of a form DF. If weighting coefficients that are automatically determined are manually adapted, this has to be done relative to already manifested weighting coefficients.

To be more concrete, this paragraph sets out how to determine the weighting coefficients for the parameters of one object. A row in figure 5.38 corresponds to a data set and a data set here is equivalent to an *1:n* relation between an object of the product and objects of the process model. Data sets with an *m:n* relation are excluded since they distort the result. To determine the weighting coefficients for the parameters of one product model object, only the portion of data sets which has at least one parameter value for a parameter of that object is applied. These data sets are marked with an arrow in figure 5.38 with all others ignored. The number of cost-parameter diagrams that are generated corresponds to the number of parameters which the object has. The column that contains the total costs of an operation has a downward diagonal pattern. In the diagram on the left-hand side, the values of product parameter P_1 of the cells with the upward diagonal pattern and the total operation costs are applied to generate the data points. For the second diagram, the cells with the upward diagonal pattern of product parameter P_n and the total operation costs are applied. Analogously, the weighting coefficients can be determined from the dot clouds of each diagram. If one of the parameters is of type nominal, the determination of the parameter m_j has to be done according to figure 5.37. The fact that a parameter of an object depends on another is of no importance for the determination of the weighting coefficients.

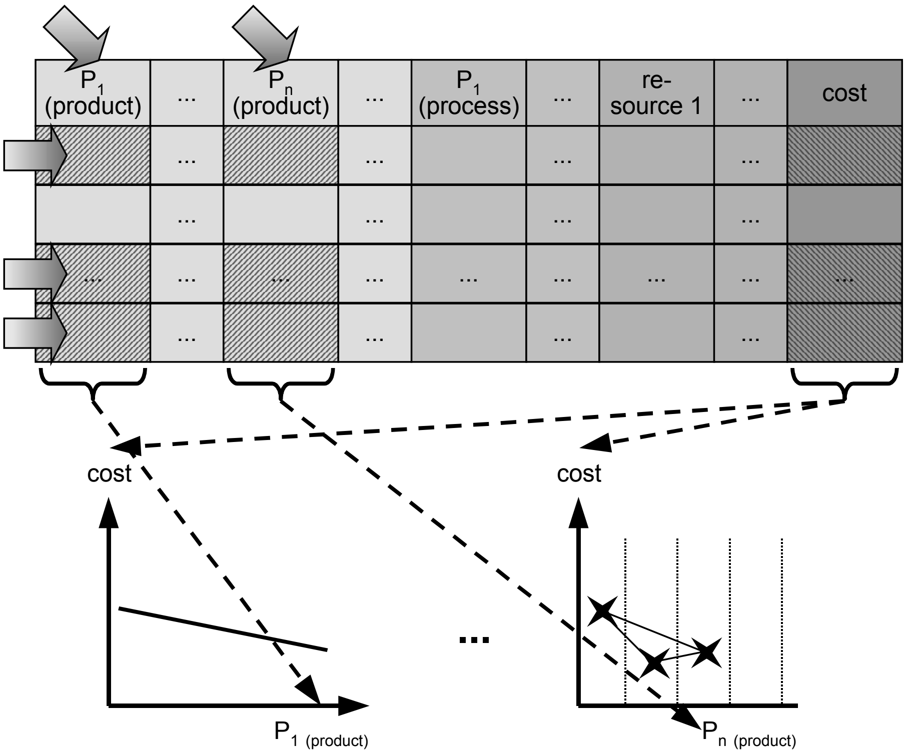

Figure 5.38: Weighting Coefficients for Parameters of One Product Model Object.

The same data sets as used for the determination of the weighting coefficients for the parameters are applied to determine the weighting coefficients for the various criteria with regard to which the allocation functions of a parameter are defined (provided that the product object is identical). Yet not product but process parameters, process classes, and resource classes and instances are applied. The other steps set out above are carried out in the same way. The weighting coefficients for the various criteria are the same, irrespective of the parameter of the object for which they are applied. Figure 5.39 depicts how these weighting coefficients for the various criteria are worked out. Every diagram generated is valid for an object of the product model and represents the weighting coefficient for one criterion, e.g. process parameter P_1.

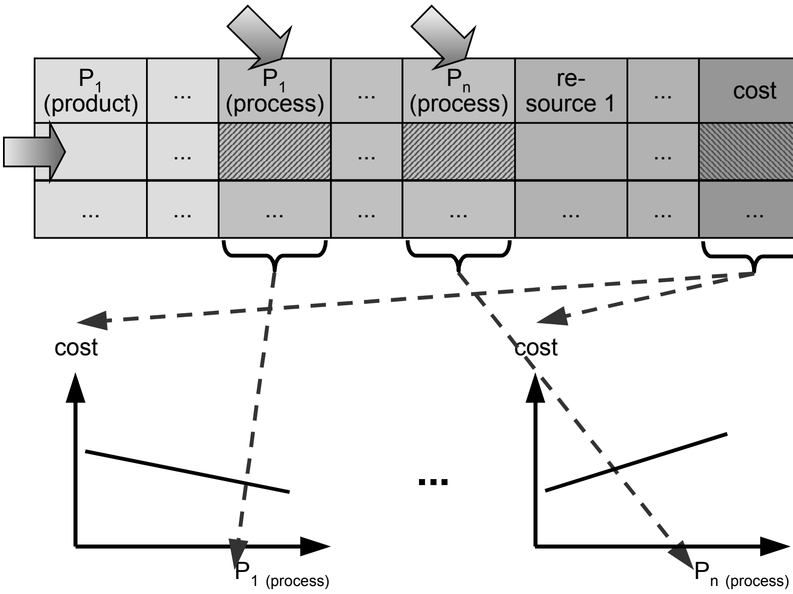

Figure 5.39: Weighting Coefficients for Criteria of One Object.

In contrast to the manual definition of the weighting coefficients for object similarities in a compound (see figure 5.26), weighting coefficients for a form DF and its technological DFs are automatically determined. Each parameter out of the set of form DF and technological DF parameters has a weighting coefficient that weights the parameters within one DF against each other and in addition against the parameters of the other DFs in the compound. To do so, also only the data sets with *1:n* relations are applied. For every form DF and technological DF parameter, the weighting coefficient is computed as described above, applying equation (13) with *j* now indicating the parameter *j* out of the set of the form DF and its interrelated technological DFs. Thus, the weighting coefficients for interrelated objects are set to 1.

In addition, for the parameters and criteria that are non-negligible because weighting coefficient determination showed that they greatly impact resource consumption, the allocation functions (numerical parameter) and similarity matrices (nominal parameter) have to be determined for the instantiable classes (use of specific knowledge presupposed). Thus, again a distinction has to be made between numerical and nominal parameters. A possible criterion is one of the following:

- A parameter of an operation applied to manufacture the instantiable product model object class, e.g. feed rate.
- An abstract operation class.
- An abstract resource class.
- An instantiable resource class.

Even though the weighting coefficients for the criteria of a product model object are the same, the allocation functions are usually not identical.

For the determination of allocation functions and similarity matrices, the data resource can be subdivided into a definition space and a solution space. The former corresponds to the objects and parameters of the product partial model and the latter to objects and parameters of the process model and to objects of the resource model.

Data sets in the data resource that are applied to find the allocation function for an object parameter with regard to a defined criterion have a parameter value for an object parameter of the definition space and a parameter value for an object parameter of the solution space. The vertical axis of the diagram as depicted in figure 5.40 corresponds to an item of the solution space; the horizontal axis always corresponds to parameter of an object of the definition space. Each data point originates from another data set. The data points have to be clustered. Such a diagram is to be generated for every combination of objects of the definition and the solution space with respect to possible criteria as depicted above.

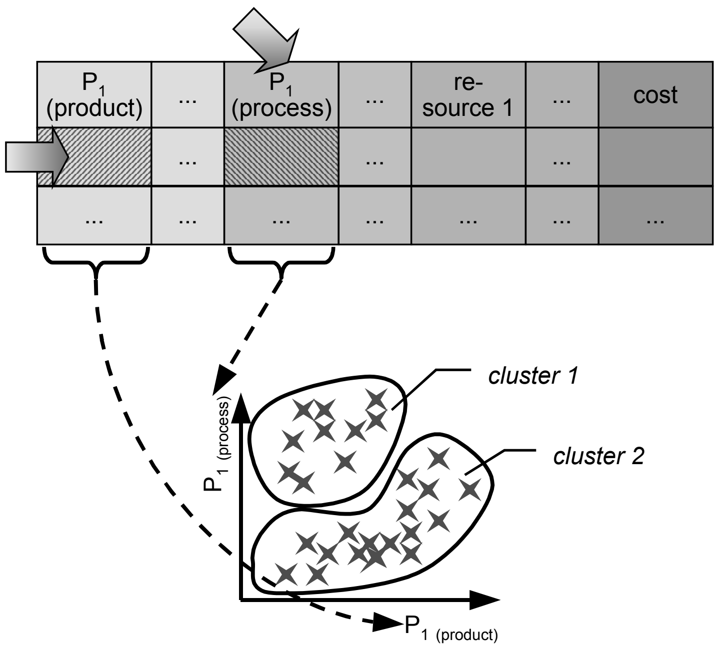

Figure 5.40: Dot Cloud and Cluster.

The objective is to provide some information on this set of data. Hence, additional information concerning the product model object is applied, i.e. the other parameter values of the product model object. With this information, the points are classified into groups with as little differences as possible. The variance of each homogeneous group is maximum (external heterogeneity), and the variance in each group is minimum (internal homogeneity). Whenever objects are to be clustered, it is necessary to define criteria for which the clustering is to be carried out.

For cluster analysis, objects with *n* parameters are represented in n-dimensional vector space. The relative position between the points represents their proximity. Hence, the product model object with *n* parameters for which the allocation function for parameter *n=1* is to be determined is represented in an (n-1)-dimensional vector space (the missing dimension is the parameter *n=1*). *n* is the subset of parameters of the set of *m* product parameters that belong to one product model object. In order to be able to represent the parameters with different parameter types (binary, nominal, numerical) and different units in this (n-1)-dimensional vector space, one of the following methods has to be carried out: level regression, level progression, or weighted assessment.

This thesis proposes to apply level progression, i.e. the parameters of the object are transformed in a unified representation on a higher level (numerical). First, suitable clusters have to be determined in the progressed (j-1)-dimensional vector space where *j* corresponds to *n* dimensions mentioned above and is the number of parameters of one product model object. Then, the clusters are mapped to the data points in the two-dimensional vector space as depicted in figure 5.40. Different alternatives are conceivable for generating the clusters. They differ largely in the way the proximity is determined and in the merging algorithm, e.g. hierarchical, partitioning. The results that are obtained for the different procedures vary. This thesis proposes the hierarchical-agglomerative cluster generation for the data points in the (j-1)-dimensional vector space. Hierarchical approaches in contrast to partitioning approaches take a longer computation time, which is however neglected.

The procedure for cluster generation is as follows:

- Generate the most detailed cluster, i.e. every data point is seen as a cluster with exactly one element.
- Compute the distances (Euclidean distance) between all *p* clusters.
- Identify the two clusters *a* and *b* that have the least distance.
- Generate a new cluster *c* out of the two clusters *a* and *b* and compute the median for *c*.
- Again compute the distances for the *p-1* clusters.
- Identify the two clusters that have the least distance and merge them.

Proceed with determining the distance of the *p-2* clusters and generating new clusters as long as the quality of cluster generation increases. To judge the quality of cluster generation, a power function is applied.

The following steps are performed for the clusters in figure 5.40 to determine allocation functions (see figure 5.41):

- Generate regression curve for the dot cloud of one cluster.
- Prepare the regression curve in order to make it differentiable.
- Differentiate the regression curve.
- Take the absolute value of the differentiation.
- Integrate the absolute value.
- Normalize the integral to value range from 0 to 1.

The result of this procedure is an allocation function for interacting numerical parameters in such way that the parameter of the process model object corresponds to the criterion described in chapter 5.7.2 and the parameters and their values that specify the cluster correspond to the interacting parameter introduced in chapter 5.7.3. The procedure set out above has to be repeated for every cluster.

The allocation functions generated are monotonic increasing in nature, but the degree of change is not limited to the elements in the library of ramp and step functions as set out in chapter 5.7.2. Again, the function is only valid within the definition range. For the dot clouds of every cluster the variance is computed to indicate the probability of the correctness of the regression curve.

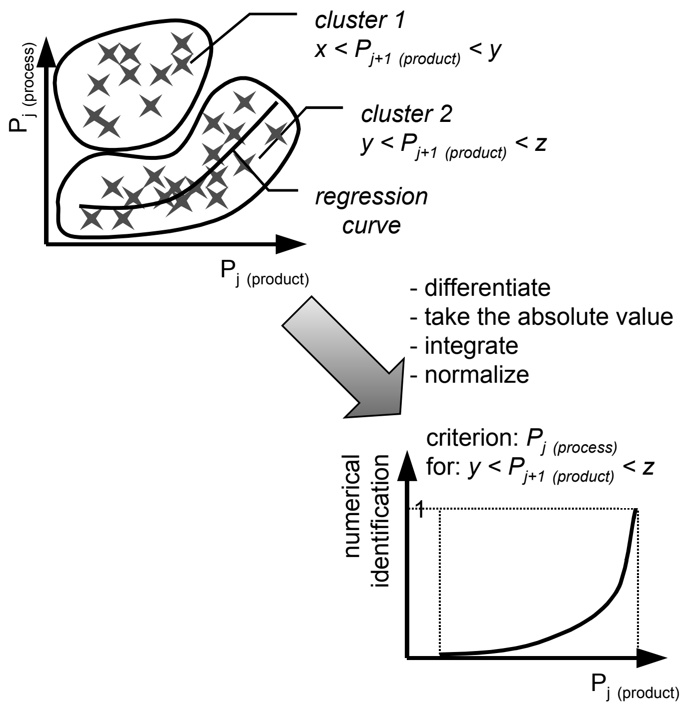

Figure 5.41: Allocation Function for one Cluster.

Figure 5.42 illustrates how to determine the allocation function for a nominal parameter from the solution space with single values in one data set. Every possible nominal parameter value generates a group on the vertical axis. A data point is generated by putting a data point in the row of the corresponding group at the position on the horizontal axis that corresponds to the numerical parameter value. Again, the attempt is made to figure out proper clusters and generate the allocation function. For this reason the nominal parameter values are replaced by integers. When computing the regression curve, these integers are assumed to be direct neighbors. The discriminating characteristic of the cluster becomes the interacting parameter and the nominal parameter the criterion.

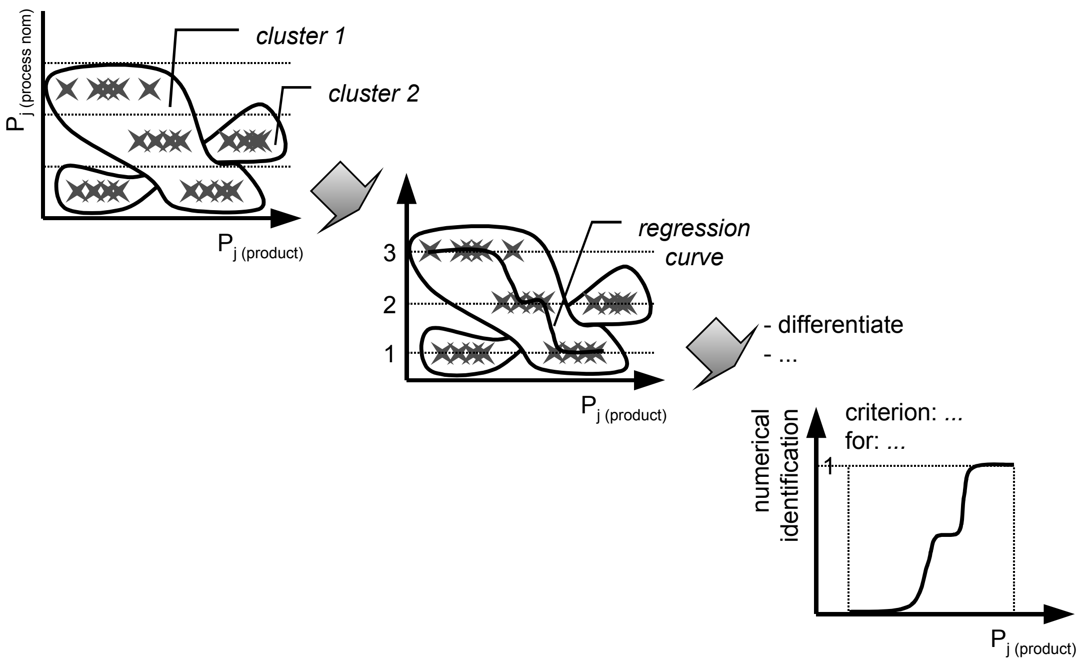

Figure 5.42: Determination of the Allocation Function for Single Values in Data Set.

Figure 5.43 illustrates the determination of the allocation function for multiple values in one data set. Multiple values occur if an abstract operation class, an abstract resource class or an instantiable resource class is chosen as future criterion of the allocation function. The multiple values in one data set are reduced to one integer. Again, the variance is computed for the regression curve for both single and multiple values.

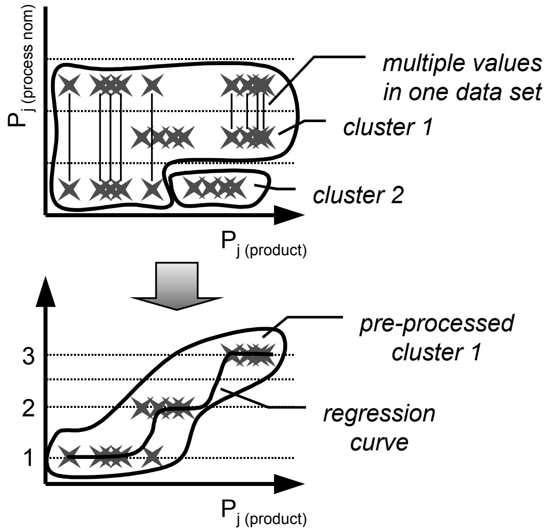

Figure 5.43: Determination of the Regression Curve for Multiple Values in a Data Set.

The determination of the allocation function with respect to resources should be carried out on the basis of resource instances. However, the resource instances have to be prepared, and instances of the same super-class possessing the same capability and the same specific cost rate are considered as one single resource instance. Furthermore, allocation functions are determined for selected abstract classes of resources, e.g. machine tool, machining tool, operating supply, auxiliary supply. For each of the abstract classes selected, the instances of these abstract classes are the groups for which the application is put on the parameter value of the parameter of the product model object. No distinction is made as to whether or not resources possess a specific cost rate. For the determination of allocation functions with regard to resources, whether clustering is necessary also needs to be checked.

Figure 5.44 depicts the determination of an allocation function for a nominal product parameter and a numerical parameter of the solution space. Clustering of the data points may be necessary. The arithmetic median is generated for every cluster. Subsequently, the topic categories are arranged with increasing arithmetic median. Then, the absolute value of each step s_j has to be computed as follows:

$$s_{j,A\,and\,B} = \frac{\left(p_{j,B} - p_{j,A}\right)}{p_{j_{\max}}} \tag{14}$$

with $p_{j,A}$ and $p_{j,B}$ being the succeeding arithmetic medians of the clusters for the product parameter value *A* and product parameter value *B* and p_{jmax} being the highest arithmetic median of a cluster for the product parameter *j*.

Only steps are possible since nominal product parameters have discrete values (see figure 5.44) and not continuous parameter values (see figure 5.42). The information contained in the allocation function could also be represented in a similarity matrix. The criterion for which the allocation holds true is the former discriminating parameter of the process model object. Again, the allocation function is applied to determine the similarity between two objects. To compute the variance, equation (12) is applied.

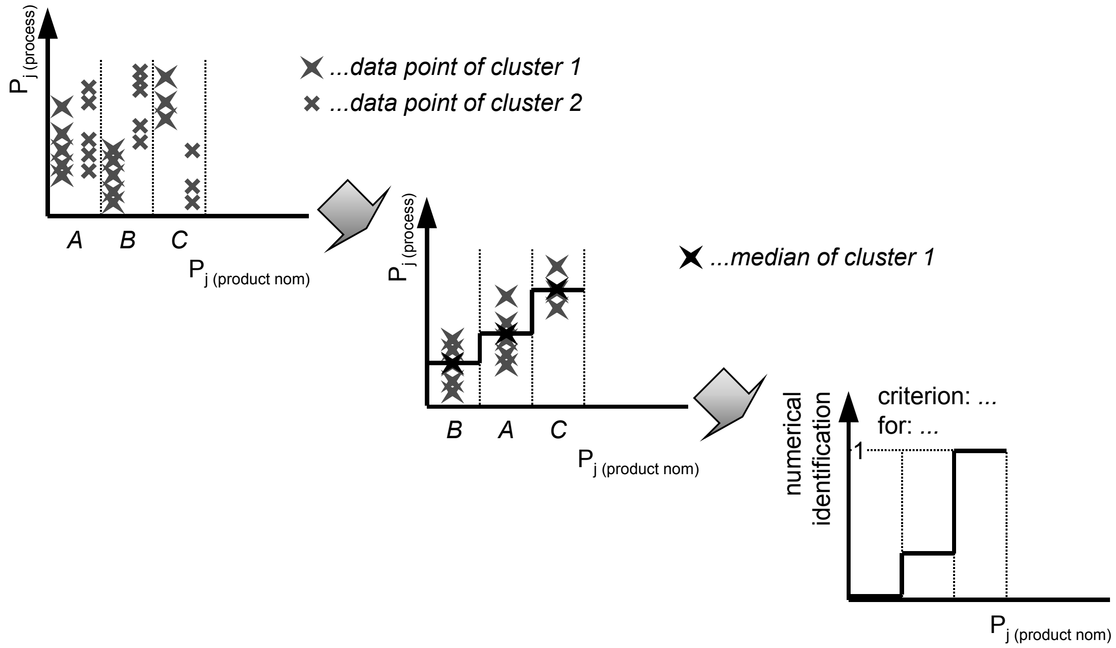

Figure 5.44: Determination of Allocation Function for Nominal Product Parameters.

A similarity matrix instead of an allocation function is to be generated if both the parameter of the definition and the parameter of the solution space are nominally scaled. Figure 5.45 illustrates how the similarity matrix can be determined. Again, only the groups of homogeneous data are applied to compute the similarity matrix. Thus, this similarity matrix may also possess interacting parameters.

The procedure is as follows:

- Generate data point in corresponding cell *kl*.
- Cluster data points.
- Sum up for each cell the number of data points in each cluster c_m (quantity $q_{kl,cm}$).
- Calculate the ratio $r_{kl,cm}$ of the distribution to nominal value classes of the solution space.

$$r_{kl,c_m} = \frac{q_{kl,c_m}}{\sum_{l=1}^{n} q_{kl,c_m}} \tag{15}$$

- Determine the absolute value of the difference $d_{kl,(k+1)l,cm}$ between the distribution ratios $r_{kl,cm}$ and $r_{(k+1)l,cm}$ of the product parameter classes.

$$d_{kl,(k+1)l,c_m} = \left| r_{kl,c_m} - r_{(k+1)l,c_m} \right| \tag{16}$$

- Aggregate the absolute values.

$$d_{k,(k+1),c_m} = \sum_{l=1}^{n} d_{kl,(k+1)l,c_m} \tag{17}$$

- Assess similarity.

$$sim_{k,(k+1),c_m} = 1 - \left(\frac{d_{k,(k+1),c_m}}{2} \right) \tag{18}$$

- Put similarity in similarity matrix.

The similarity matrix is only valid for the differentiating characteristic of the cluster *m* and the nominal parameter of the solution space becomes the criterion $c_{j(solution\ space)}$. In case that a data set has multiple values, an additional group for the nominal parameter is generated.

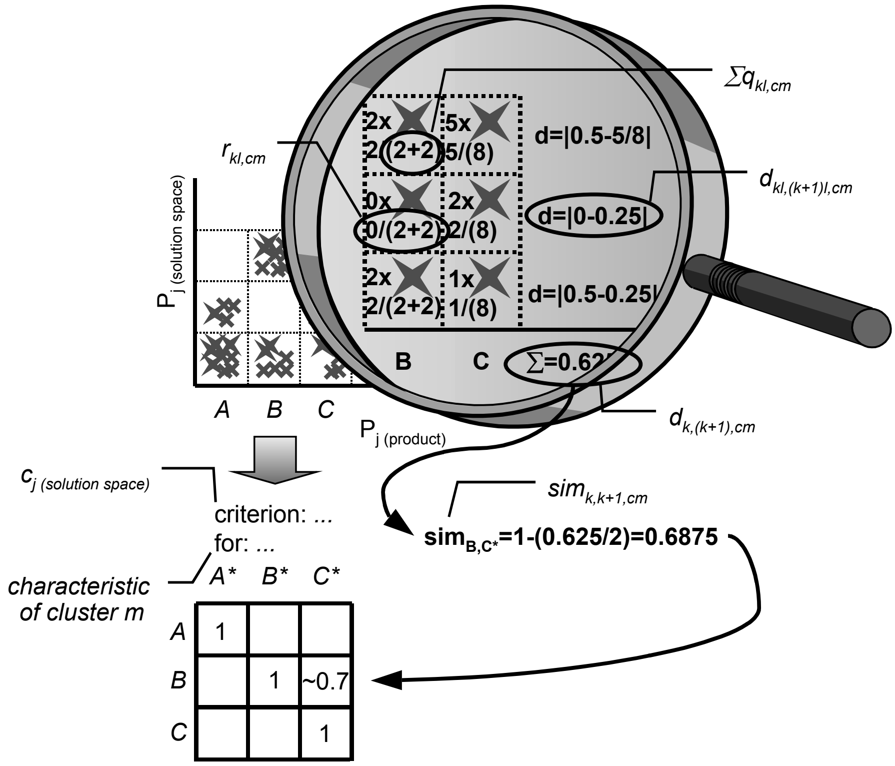

Figure 5.45: Determination of Similarity Matrix for Nominal Parameters.

This chapter has discussed different alternatives for reducing the effort for similarity definition: inheritance and copying, for example, or determining a reasonable correlation between product characteristics and manufacturing or manufacturing costs either to guide the user or to automatically detect allocation functions, similarity matrices, and weighting coefficients. Not always can every possibility be taken advantage of. If, for example, there are too few data sets with a *1:n* relation in the data resources or if the form DFs are related to technological DFs of different types, the similarity measure generally cannot be determined at all or, if determinable, not with any sufficient degree of quality. In this case, the similarity measure should be defined manually.

As previously described, CBR cannot guarantee a 100% correct solution. The exactness of solution is traded off for an approximate solution that is controlled by the similarity measure. In addition, the weighting coefficients, allocation functions, and similarity matrices automatically determined as sketched in this chapter incorporate a certain level of uncertainty, i.e. the correctness cannot be fully guaranteed. The variance that is computed for weighting coefficients and allocation functions indicates the uncertainty of their correctness and advises the manufacturing expert to validate or, if necessary, modify them. The reason for the uncertainty lies in the correlation of the parameters in the data resources. To which extent the options described can be applied to reduce the effort for similarity definition needs to be further elaborated based on real data.

5.8 Cost Estimate for Direct Manufacturing Costs

The CAD product model is made up of GEs, form DFs and technological DFs and comprises the information on the type of product. For every form DF and GE, the instantiation relations are backtracked from the CAD product model to the product partial model. The result of the backtracking of instantiation relations is the identification of every instantiable class in the product partial model that corresponds to an object of the CAD product model. For the identified instantiable classes either specific or generalized manufacturing knowledge is employed to generate the operations for this cost estimate depending on whether the class has a generative or an informational IR.

Figure 5.46 shows the instances in the CAD product model and highlights the instantiable part class as well as the instantiable classes of the form DF instances. The instantiable part class and the instantiable form DF class *a* have, in each case, a generative IR, whereas the instantiable form

DF class *b* has an informational IR.

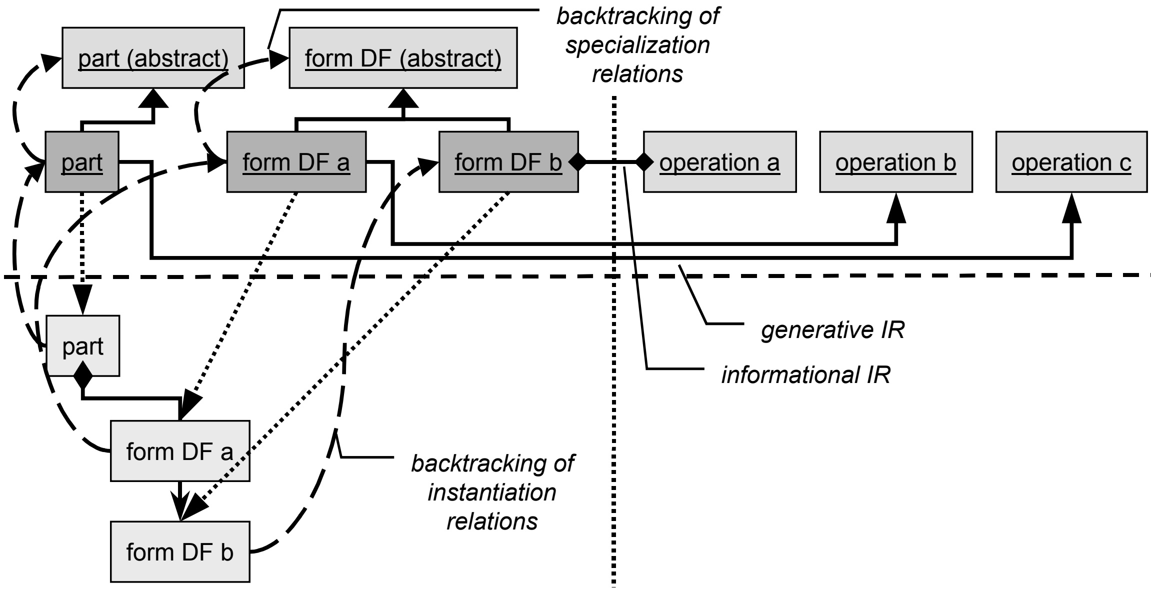

Figure 5.46: Backtracking of Instantiation and Specialization Relations.

For the instances that have an informational relation on the class level, retrieval and reuse as described in chapter 5.6 are carried out until each of these instances has suitable operations and resources. The operations are related to the instances of the CAD product model, thus primarily filling the operation pool.

For each identified instantiable class that has no informational IR, the specialization relations are backtracked. A check is made for every super-class of these sub-classes to determine whether it has IRs or not. The classes that have generative IRs are again highlighted. The same has to be done in the part taxonomy in order to determine IRs for abstract part classes.

Figure 5.46 also shows the backtracking of the specialization relations for the instantiable classes that have no informational IR and highlights the abstract form DF class *a* and the abstract part class, which in this case do not possess a generative IR.

Subsequently, the generative IRs of the identified instantiable and abstract classes and the related object classes are instantiated with regard to relation-to-part relations and exclusionary relations. Also, the generative IRs of the operations are instantiated, with the resources necessary to perform the operation selected from the resource library. The resource library allows generative IRs solely to choose a resource from a discrete number of resources, not to instantiate a new resource. Not only concurrency and structuring relations but also the related structuring elements are instantiated, thus expanding the pool of operations.

Figure 5.47 portrays the instantiation of two operations. In addition to the reused operation *a*, the pool of operations is expanded by the two operations *b* and *c*. The figure does not depict the IRs that start from the operations and end at resources.

To sum up, the result of the application of generalized manufacturing knowledge and the reuse of specific manufacturing knowledge is a pool of operations. However, this pool is not yet applicable for cost estimation. Rather, additional steps are necessary to structure, unify, and complete the pool.

Operations that would lead to redundant results have to be eliminated as depicted in figure 5.21. To do so, exclusionary relations between generative and informational IRs are interpreted. Then, the operations are combined to groups applying structuring relations and structuring elements such as sub-job, job, and string of jobs. The orientation of the form DFs is, for example, the discriminating characteristic used to group them into sub-jobs.

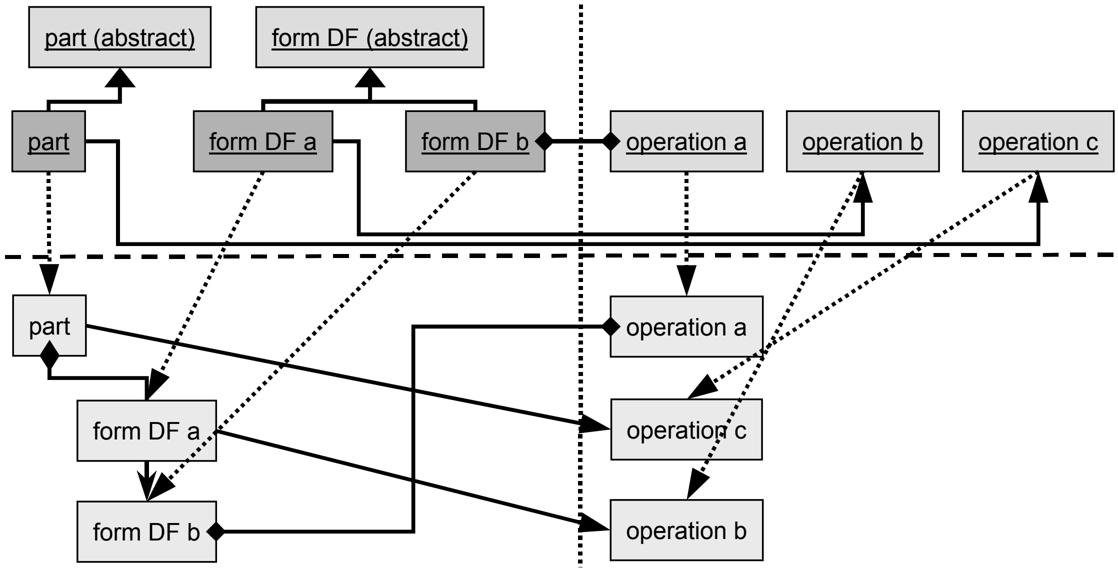

Figure 5.47: Instantiating Generative IRs.

Since the reused operations in the pool of operations originate from integrated PPR instance models of various parts, they apply a multitude of different resources, e.g. machine tools. For groups of operations the resources mostly belong to the same class or are comparable with regard to the specific cost rate. In order to avoid unnecessary machine or machining tool changes, the multitude of resources has to be reduced. Therefore, especially in the case of machine tools it is the one that is used most frequently that then replaces the others.

Some resources, especially equipment, necessitate further preparatory or post-processing operations. It is presupposed that this knowledge is explicitly available. In the integrated PPR model, a generative IR starts from a resource class and ends at an operation class. Thus, for every resource instance in the pool of operations, the instantiation relation is backtracked and the generative IRs of this resource class instantiated. In the example of figure 5.48 the resource *d* necessitates operation *d* which is added to the pool of operations. Some of the operations can only be instantiated once within a structuring element, e.g. setting up the part. In contrast others can be instantiated multiple times, e.g. positioning movement of the machining tool. This again allows operations to be added to the pool of operations, which is now complete.

The result is an integrated PPR instance model that comprises both the operations necessary to manufacture the CAD product model and the required resources. It quantifies the degree of resource consumption per operation. Subsequently the single costs of the operations are aggregated in order to obtain the direct manufacturing costs.

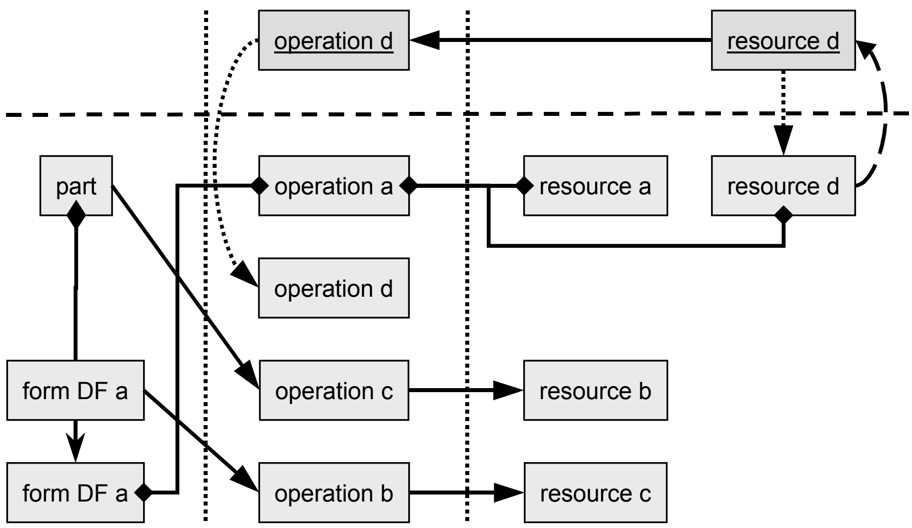

Figure 5.48: Adding Resource-specific Operations.

The costs are determined on a very detailed level and have a relation to the objects of the CAD product model. Therefore either a design or a process-oriented cost structure, in each case at different levels of granularity, is possible.

To handle explicit and tacit manufacturing knowledge is the challenge for estimating direct manufacturing costs based on the integration of product, process, and resource. Thus, chapter 5.5 has described representation of generalized manufacturing knowledge and detailed the (re)use of specific manufacturing knowledge. How to manually define and to automatically determine manufacturing similarity has therefore been elaborated. Different alternatives for reducing the effort for similarity definition have to be validated on the basis of real data.

5.9 A Recapitulation on the Cost Estimation Model for Direct Manufacturing Costs

This chapter intends to recapitulate on the cost estimation model for direct manufacturing costs and on how the modules developed interact. Figure 5.49 depicts the modules, distinguishing between the preparation and the application side. The latter is concerned with the process of estimating the direct manufacturing costs for a certain product model whereas the former stores and provides the information and knowledge necessary for the cost estimation process either at run-time or in advance.

The PPR classes and the generalized manufacturing knowledge, i.e. generative IRs, relation-to-part relations etc., are stored in the integrated PPR class model. Episodic cases and prototypical cases, both of which incorporate specific manufacturing knowledge, are stored in the case base. Allocation functions, similarity matrices, and weighting coefficients, regardless of whether manually defined or automatically determined, are represented in the similarity measure database. Solely for reasons of clearness, the databases are separated. Implementing the cost estimation model does not necessitate physically independent databases.

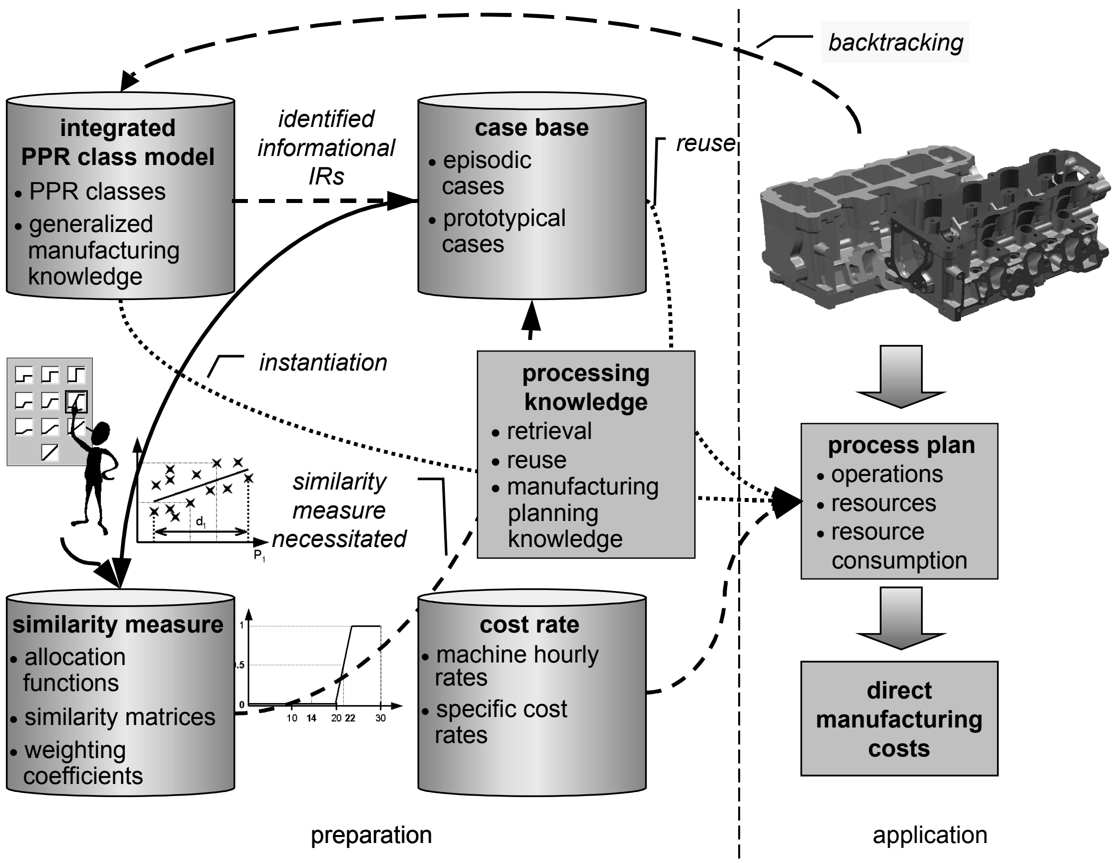

Figure 5.49: Cost Estimation Model for Direct Manufacturing Costs.

For cost estimation, the instantiation relations of the objects of the finished and the rough part model are backtracked in order to identify the corresonding PPR classes, either instantiable or abstract. For classes of the product partial model with generative IRs, the corresponding operations are instantiated. For classes with informational IRs, the processing knowledge for retrieve and reuse is applied in order to identify the similarity measure related to this class and to find the most similar case in the case base. Manufacturing planning knowledge is applied to carry out activities such as grouping of operations and unifying machine tools. The cost rates are finally applied to monetary value the newly computed resource consumption of each operation.

PART III: Application of CABACO

Chapter 6

Implementation of CABACO

6.1 Prototypical Implementation

To validate the cost estimation model, the prototype cost estimation system CABACO is developed. CABACO is the acronym for ***ca**se-**ba**sed **co**st estimation* and is focused on that part of the model for cost estimation in detail design that relies on specific manufacturing knowledge. The purpose is to test the practical applicability of the retrieve and reuse step as described in the chapters 5.6.1 and 5.6.2 as well as the definition of knowledge about manufacturing similarity (chapters 5.7.1 to 5.7.4) and the definition of prototypical cases.

The representation and application of generalized manufacturing knowledge as described in chapter 5.5 is currently under development. The computation of overheads as described in chapter 4.4 is only partially implemented. The possibilities to decrease effort for similarity definition, that are described in chapter 5.7.5, and the determination of prices for rough parts and external services as described in chapter 4.2 are not implemented yet. Figure 6.1 indicates the position of the CABACO prototype implementation. The limitation to this part of functionality has nothing to do with general validity of the model for cost estimation which is indisputable but is driven by the fact that human resource implementing capacity has been focused on the most crucial part of the model.

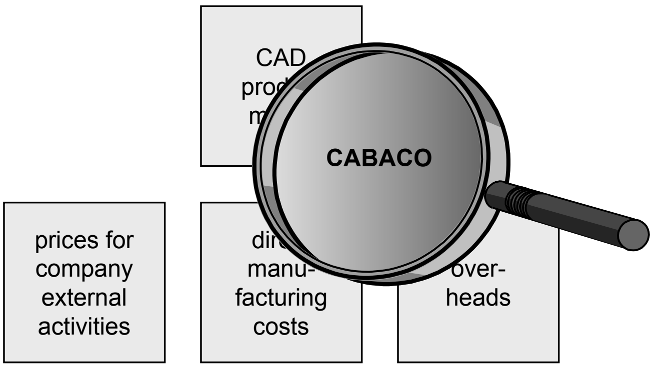

Figure 6.1: The Position of the CABACO Prototype Implementation.

The CABACO system has been developed using Microsoft Visual Basic. The system structure allows the continuous creation, deletion, and modification of specific manufacturing knowledge, knowledge about manufacturing similarity, object classes, and resource instances. The implementation of CABACO incorporates the processing knowledge, functionality for the interpretation of the CAD product model, the database, and the spreadsheet calculation. At present, CABACO comprises only processing knowledge: thus far no domain knowledge has been included. Domain experts have to represent the latter in the CABACO database, which can be done without any programming effort because CABACO provides a simple user interface.

The CAD system used is CATIA V5, Microsoft Access is chosen as database and Microsoft Excel is the spreadsheet calculation. Microsoft Visual Basic 6.0 is the programming language. Figure 6.2 depicts the architecture of CABACO. A CATscript macro in CATIA V5 saves the CAD product model in ASCII code, which is subsequently transferred to Microsoft Excel. CABACO controls the access to the Microsoft Access database and the Microsoft Excel spreadsheet calculation. The generated cost report is displayed in CABACO and saved as an HTML file.

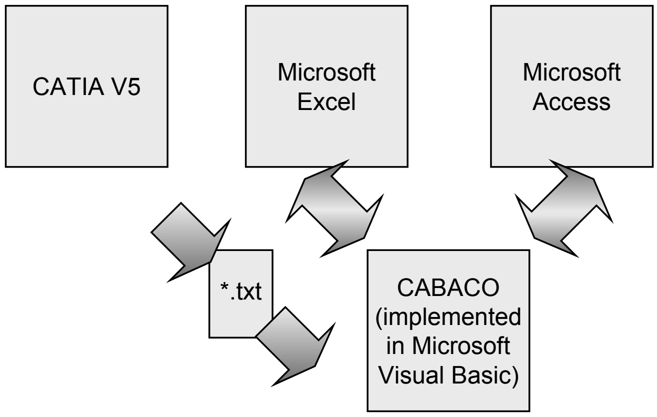

Figure 6.2: CABACO Architecture.

6.2 CABACO Use Cases

To describe the functionality and external system behavior of CABACO a use case diagram is developed and the use cases of this diagram are subsequently further detailed. Usually, use case diagrams are applied to determine user requirements; here they are applied to describe scope of functions. Use case diagrams depict actors, use cases and their relations. An actor acts exterior to the system and is involved in the interaction with the system that is described in the use case. An actor is either a user or a system.

Figure 6.3 depicts four actors: the design engineer, the manufacturing engineer, the controller, and the CAD system CATIA V5. In the use case diagram, Microsoft Excel and Microsoft Access are not considered as external to CABACO and are therefore not depicted as actors here.

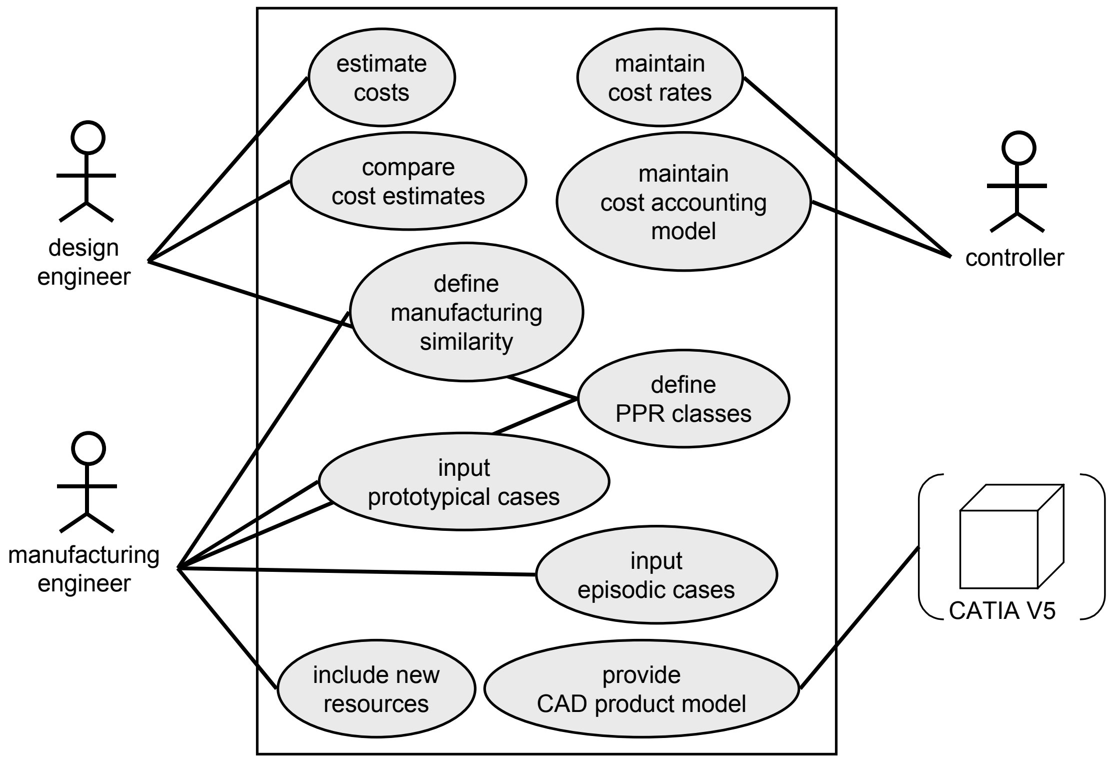

Figure 6.3: CABACO Use Cases.

Figure 6.4 shows the position of the use cases in the integrated PPR model. The use case 'define PPR classes' adds instantiable classes either to the product, the process or the resource partial model, which is a table of the database. The use case 'include new resources' puts resource instances to the resource partial instance model and the use case 'maintain cost rates' takes resource instances for modification. 'input prototypical cases' and 'input episodic cases' adds product and process instances to the product respectively process partial instance model and chooses resource instances from the resource partial instance model. The database of CABACO has in each case a table for abstract classes, instantiable classes and instances. An instance corresponds to a data set in the instance table, and cases are generated by establishing relations between the data sets.

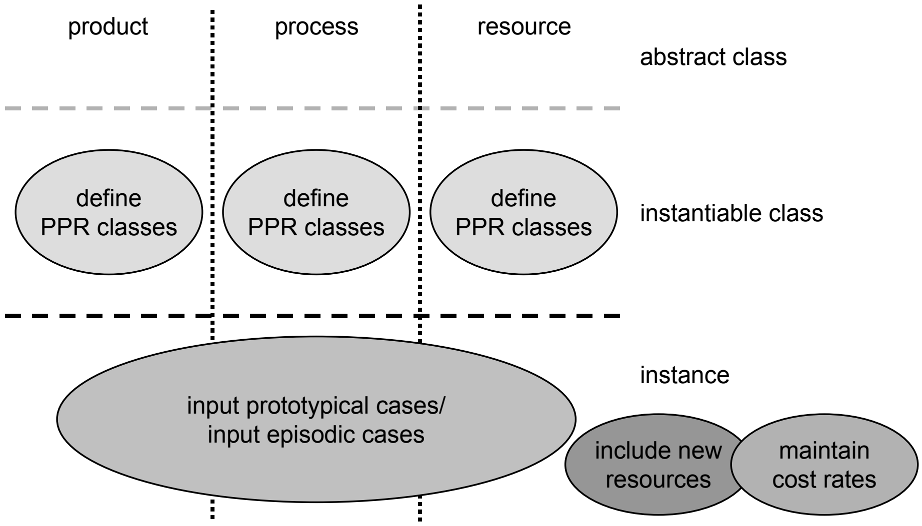

Figure 6.4: Position of the Use Cases in the Integrated PPR Model.

The abstract classes have been defined in advance in the database. This is possible if limiting the application domain to detail design and the functionality to the use of specific manufacturing knowledge. To represent and process generalized knowledge in CABACO, the user interfaces and the processing knowledge have to be completed. The database schema needs no modification.

The use case 'estimate costs' is concerned with the estimation of the manufacturing costs based on the CAD product model from CATIA V5, more precisely, with the subset of the manufacturing costs that is derived based on the usage of specific manufacturing knowledge. The design engineer is the actor of this use case that results in a detailed and differentiated cost structure. The design engineer only starts the process of cost estimation; apart from that the process is run through without any human interaction. It comprises the following steps:

- Get and interpret the CAD product model.
- Retrieve for each form DF (and its technological DFs) the most similar case.
- Reuse the *n* operations of each most similar case in the context of the corresponding form DF.
- Compute the resource consumption.
- Monetary value the resource consumption.
- Aggregate the single costs.
- Save and depict the result.

Since it is crucial to base design decisions on well-founded cost information, the use case 'compare cost estimates' is concerned with displaying the results of previous cost estimations. The design engineer selects which cost estimates to display and compares them manually.

The use case 'define manufacturing similarity' is concerned with the definition of manufacturing similarity. The manufacturing engineer is the actor for this use case and defines allocation functions for continuous numerical parameters and similarity matrices for discrete numerical and nominally-scaled parameters of a DF. Additionally, the actor determines the weighting coefficients for the criteria and the weighting coefficients for the parameter of a DF. To each data set of manufacturing similarity that is stored in the database a time stamp and user stamp is added.

Figure 6.5 depicts the CABACO's user interface to define an allocation function for a continuous, numerical parameter as part of a DF.

To define the manufacturing similarity for a DF, the manufacturing engineer has to carry out the following steps:

- Select the instantiable DF class.
  Yet, abstract DF classes can not be selected. Thus, inheritance of similarity is not possible.
- Select DF parameter.
  A parameter out of the set of parameter that are valid for this instantiable DF class can be chosen.
- Select the criterion.
  The criterion with regard to which the allocation function is defined can be selected or newly defined.
- Define weighting coefficient for the criterion.
  The coefficient is applied to weight the criteria of one parameter. They should be the same for one DF.
- Define interacting parameters and their parameter values.
  At most two interacting parameters out of the set of DF parameters can be selected and their parameter values determined.
- Determine definition range of the allocation function.
- Determine ramp and step functions.
  The manufacturing engineer can choose ramp and step functions (maximum seven) out of a library of ten and specify their position within the definition range. The functions indicate the absolute value of a variation with regard to the selected criterion. Since not every combination of functions is computable, traffic lights indicate violations.
- Save allocation function.

Subsequently, the procedure has to be repeated for the various criteria of every DF parameter.

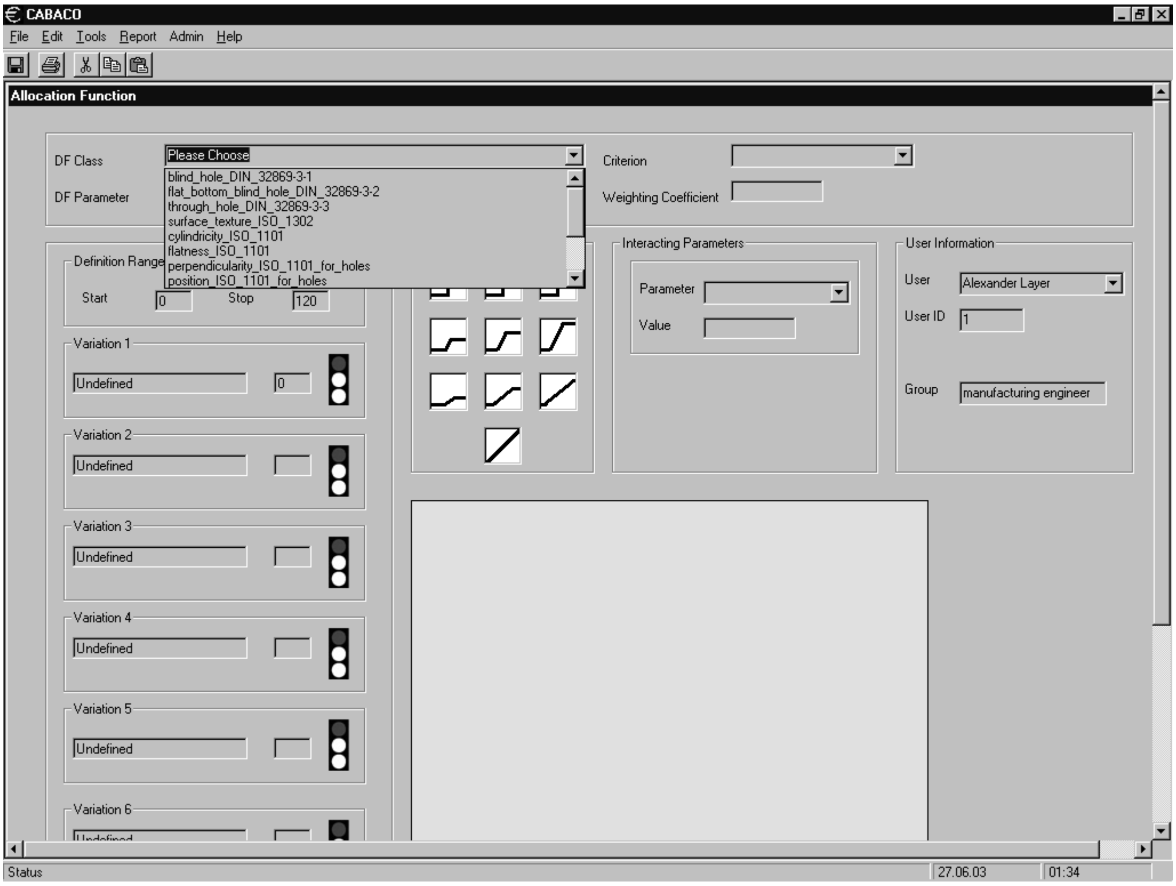

Figure 6.5: Definition of Allocation Function.

The manufacturing engineer executes 'input prototypical case'. Figure 6.6 shows the window to define a new or modify an existing prototypical case, including the actor to carry out the following steps:

- Select form DF and specify the parameter values.
- Select GEs and technological DFs and specify parameter values for the latter.
- Select MFs and specify parameter values.
- Select resources for every MF.
- Save prototypical case.

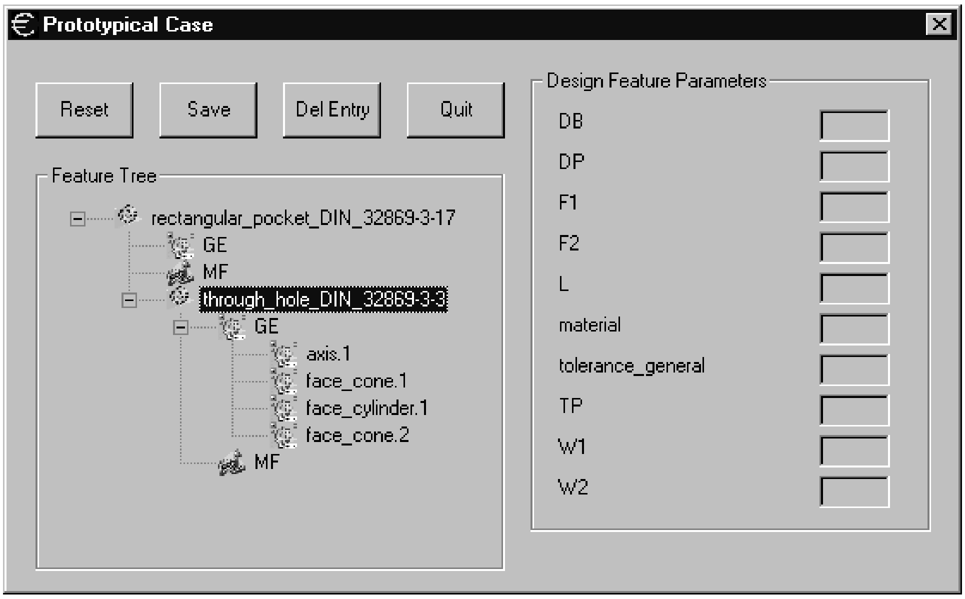

Figure 6.6: Definition of a Prototypical Case.

For the use case 'input episodic cases', the same procedure is performed. The only difference is that the prototypical case is marked as such. This bypass of simulating episodic cases is applied because it currently seems not possible to get episodic cases directly from an integrated CAx system.

A CATscript macro in CATIA V5 is applied to perform the use case 'provide CAD product model'. The macro generates an ASCII code. Since the CATscript macro is currently not able to export geometric tolerances and surface texture, the CAD product model is manually completed according to the tolerance and surface texture specification in the CAD model.

Figure 6.7 shows the user interface to define from scratch or modify an existing product class as part of the use case 'define PPR classes'. The windows for the definition of a new process and resource class look similar. They do not have the right part that defines the GEs. Instead the window for the resource classes offers to choose the unit of the specific cost rate.

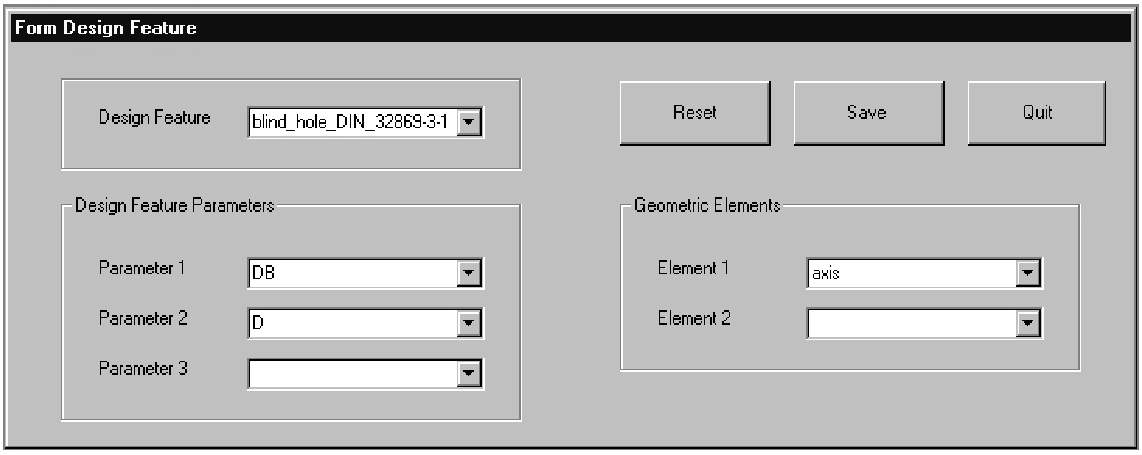

Figure 6.7: Definition of a New Product Class.

In addition to the definition of a resource class, 'include new resources' defines an instance of a resource. This includes the specific cost rate. The use case 'maintain cost rates' modifies the cost rates of the resources to keep them up-to-date. The controller performs both this use case and the use case 'maintain cost accounting model', the objective of which is to maintain the overhead rates. Of course, future implementation work targeting a productive system must focus on integrating the cost estimation application with an enterprise resource planning (ERP) system, e.g. SAP R/3, in order to avoid redundancy and to ensure consistency.

Chapter 7

Illustration: CABACO in the Metal-Cutting Domain

7.1 Application

The prototype cost estimation system CABACO focuses on that part of the cost estimation model which relies on specific manufacturing knowledge. In this chapter, CABACO will be demonstrated by means of a very simple example: A pre-cast rectangular pocket with through hole nested on the pocket's bottom. This illustration, where CABACO is applied in the metal-cutting domain, is primarily done to gain an initial impression of the applicability of the case-based cost estimation concept and to determine further insight into the requirements to be met for a productive application. The example shows the definition of product, process, and resource classes, the specification of resource instances, the representation of manufacturing knowledge as prototypical cases and the definition of the similarity measure. Finally, it illustrates the practical applicability of the retrieve and reuse step.

As has been stated, CABACO has only processing knowledge, no domain knowledge. The latter has to be represented by domain experts and is therefore part of this example. Thus, in the first step, the form DF classes for the pre-cast rectangular pocket (rough part form DF), the rectangular pocket (finished part form DF), and the through hole have to be defined. Figure 7.1 shows the definition of the form DF class through hole DIN 32869-3-3–2002. In addition to the DIN standard, the form DF class is extended with the two parameters material and general tolerance. Furthermore, the geometric elements (axis, cones, cylinder) that make up the DF are defined. Subsequently, the same has to be repeated for the pre-cast rectangular pocket and the rectangular pocket. The form DF definition does not contain the position of the DF since this is solely of interest for the form DF instance of the CAD product model for which the costs are to be determined but not for the form DF instance in the case base when assessing the manufacturing similarity.

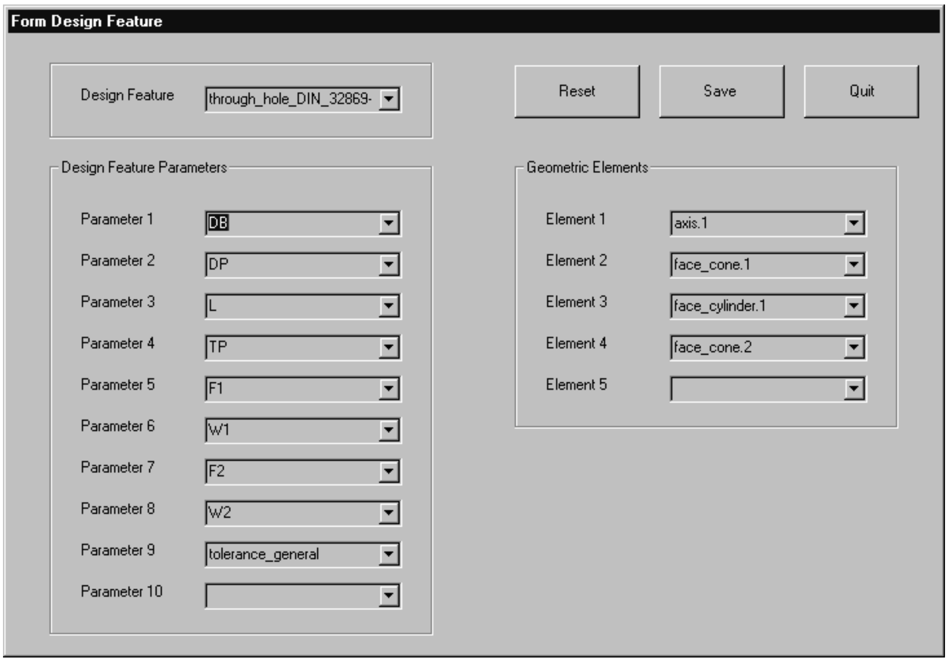

Figure 7.1: Form DF Through Hole DIN 32869-3-3–2002 Definition.

In the second step, the technological DF classes necessary to technologically specify these three form DF classes are specified. Figure 7.2 portrays the definition of the technological DF class cylindricity tolerance ISO 1101–1983. This tolerance consists of one parameter: distance of the tolerance zone. Again, the procedure for defining classes is repeated for further tolerances, e.g. flatness, cylindricity, position.

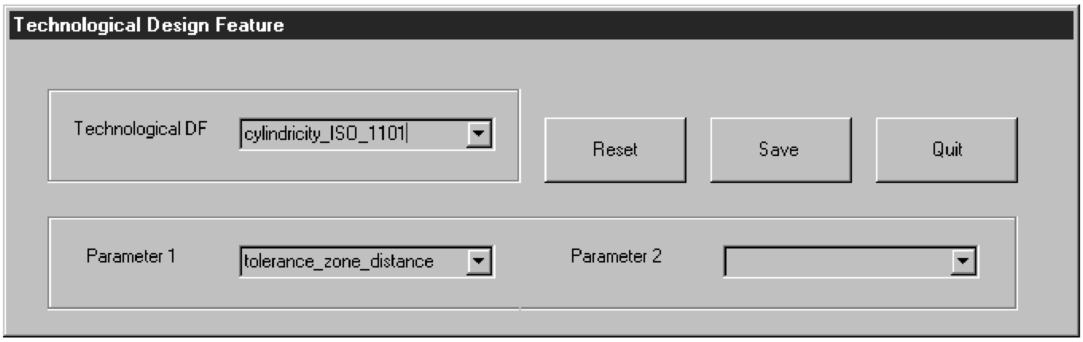

Figure 7.2: Technological DF Cylindricity Tolerance ISO 1101–1983 Definition.

In addition, surface texture indication may be necessary to technologically specify the rectangular pocket or the through hole. Thus, a surface texture DF is defined. The definition of the surface texture DF ISO 1302–2002 is done in the same window as the definition of the geometric tolerance DF classes. For the surface texture, further elements are specified: the type of manufacturing process, the manufacturing process, the surface lay, the specification limit 1, the filter type, etc. Each of the newly defined DF classes is stored to the CABACO database.

Not only the DF classes are defined but also the operation classes that are necessary to the pre-cast pocket and the through hole. Figure 7.3 depicts the definition of the MF class drilling, which consists of geometric, technological, and organizational parameters. In addition to this definition, the computation of the resource consumption and the costs of the drilling operation are added in the Microsoft Excel spreadsheet. Again, this procedure is carried out for further metal-cutting MF classes: chamfering, reaming, rough pocket milling, rework pocket milling, and finish pocket milling.

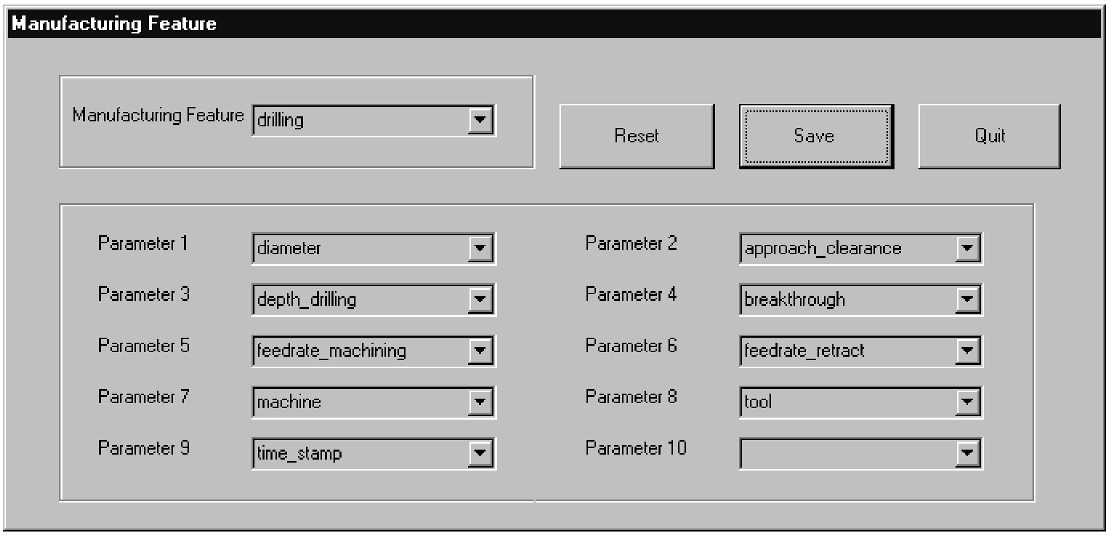

Figure 7.3: Definition of Manufacturing Feature Drilling.

To complete the definition of the classes required, resource classes – machine tools and machining tools – are added to the database. As figure 7.4 depicts, for the definition of machine tools, capability parameters are left open with only a cost rate being defined. This is done so in this illustration for a key reason: currently the revise step does not check whether a machine tool is applicable for manufacturing of a certain part. For the tools employed (twist drill, spot drill, countersink, reamer, end mill, ball-end mill, etc.), no cost rates are defined: only the nominal diameter as the subset of capability parameters that drives resource consumption is specified.

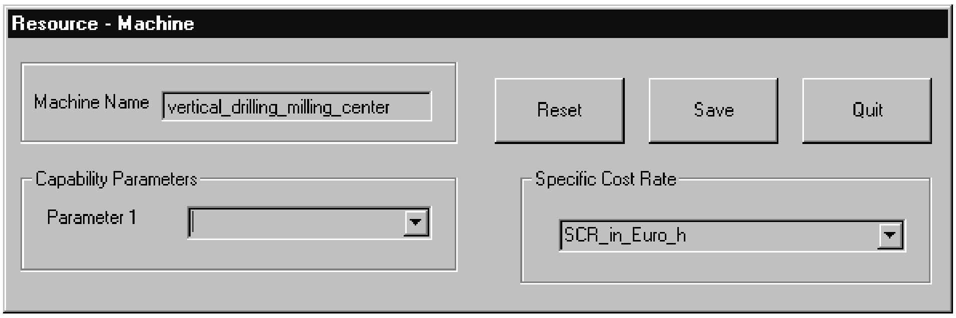

Figure 7.4: Definition of Resource Class.

Then, instances for the machine tool and machining tool classes are generated and the parameter values for the capability parameters and the cost rate are defined. Examples of instances are Danusys 1507, drill D9.8, spot drill D15, countersink D16, reamer D10, end mill D10, ball-end mill D10, etc.

The next step in this example is to define the manufacturing similarity for the form and technological DF classes. For the numerical parameters, e.g. diameter of the through hole DIN 32869-3-3–2002, an allocation as depicted in figure 7.5 is defined.

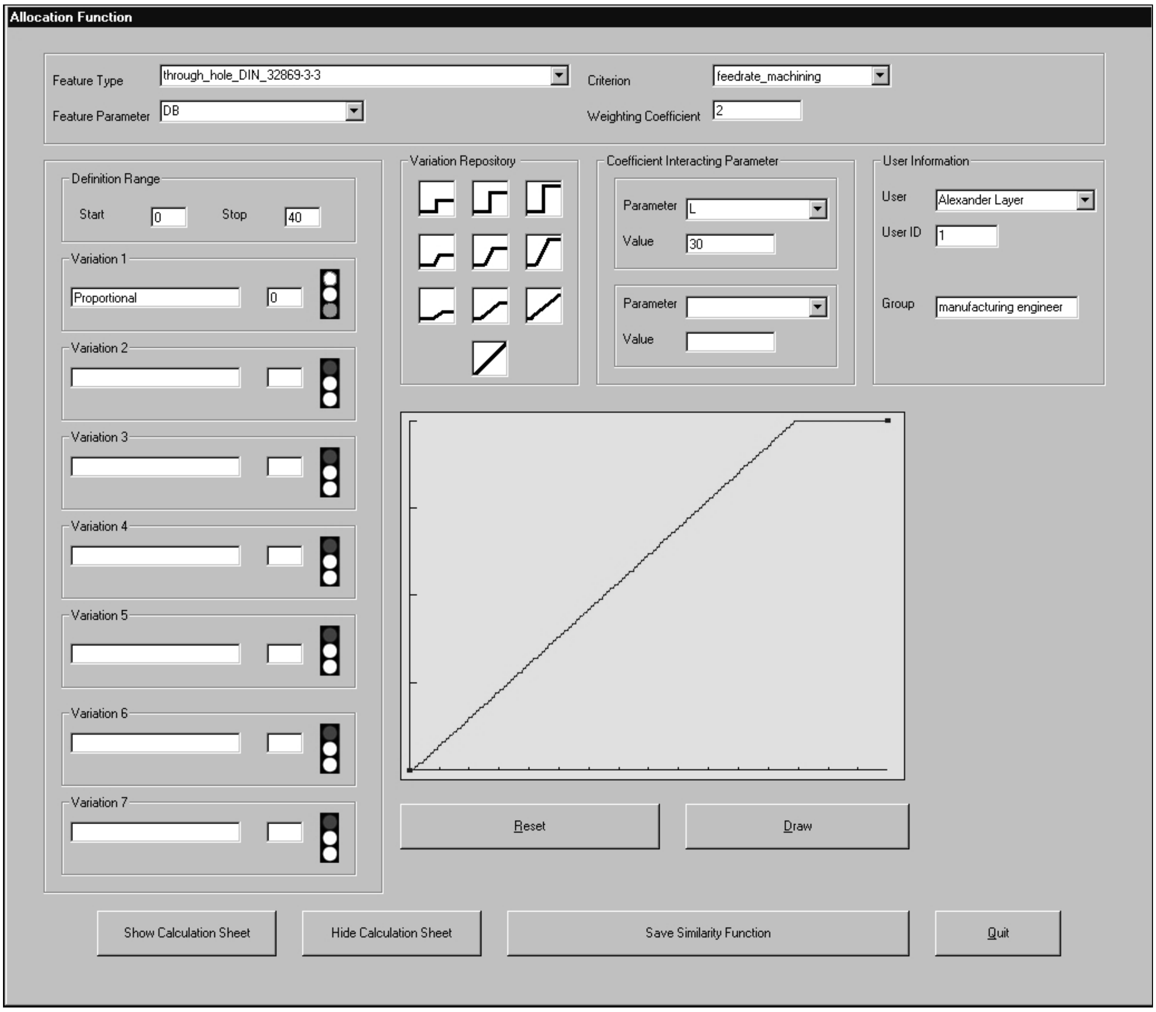

Figure 7.5: Allocation Function for Through Hole DF.

Next similarity is defined for the nominal parameters such as the material as is depicted in figure 7.6. Semantic differentials are applied to quantify the degree of manufacturing similarity between two materials with respect to the criterion selected. In the example shown below, the manufacturing similarity between *C45* and *St52* is judged as dissimilar with respect to feed rate and is weighted double. The definition of similarity applying either an allocation function or a similarity matrix is repeated for the most important parameters of the form DFs (e.g. nominal diameter of hole, diameter of fit, depth of hole, depth of chamfer 1; length, width, depth, and draft of rectangular pocket; depth of pre-cast pocket) and the technological DFs (distance of tolerance zone) with respect to criteria of significance. These criteria are weighted. To complete the definition of the manufacturing similarity for these objects, not only the parameters of each form and technological DF are weighted (depth of hole four times as important as depth of chamfer 1) but also the technological DFs that specify the three form DFs (similarity between the through hole form DF twice as important as the similarity between the cylindricity tolerance and the position tolerance).

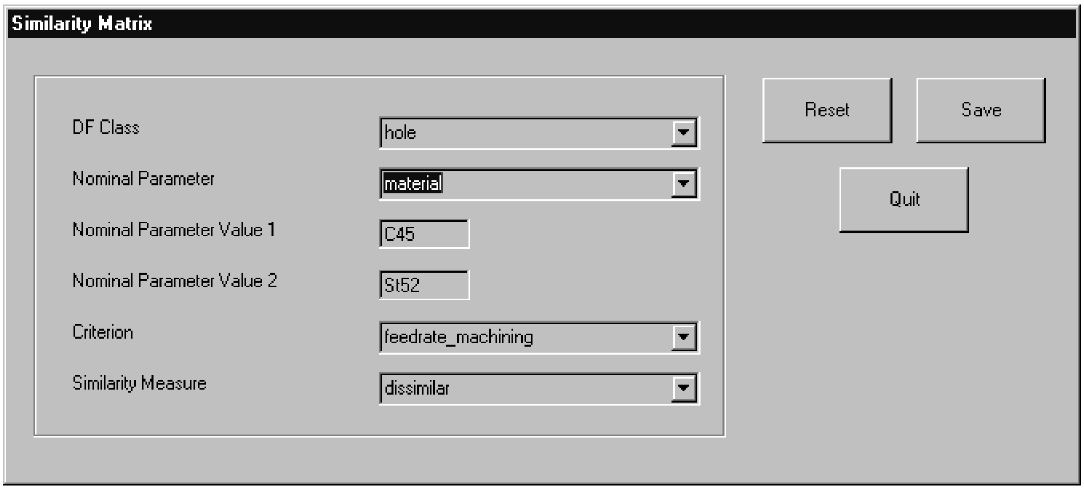

Figure 7.6: Definition of Similarity for Nominal Through Hole Parameter Material.

The result of the activities described above is that the PPR classes, the resource instances, and the knowledge about the manufacturing similarity for the DFs selected are represented in the CABACO database. In the next step, specific manufacturing knowledge is added to the database or, to be more precise, to the case base. Figure 7.7 depicts the pre-cast rectangular pocket DF, the rectangular pocket DF, and the through hole DF of the prototypical case and figure 7.8 shows the definition of this prototypical case. The parameter values of the form DFs are specified. The through hole has a position tolerance the datum elements of which are not elements of the pocket but faces outside the pocket. The distance of the tolerance zone is 0.2. The sequence of operations is rough pocket milling, finish pocket milling, drilling, chamfering, and reaming. For each of these operations the manufacturing parameters and the resources are specified.

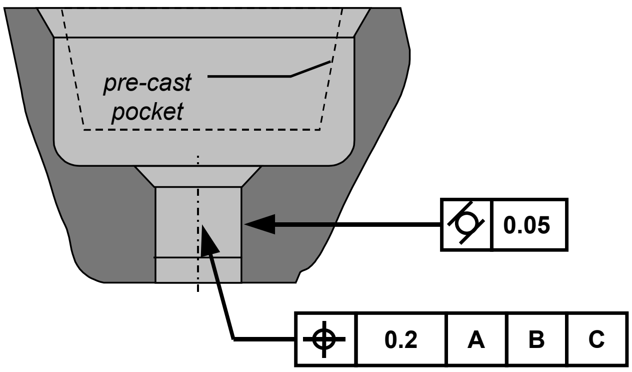

Figure 7.7: Prototypical Case.

Figure 7.8 shows the parameters and the resources – Danusys 1507 and twist drill D22 – applied to carry out these operations. This comprehensively specified case is stored as a prototypical case in the case base of CABACO. This procedure is repeated for the same form DF types but with different parameter values in order to obtain more cases in the case base.

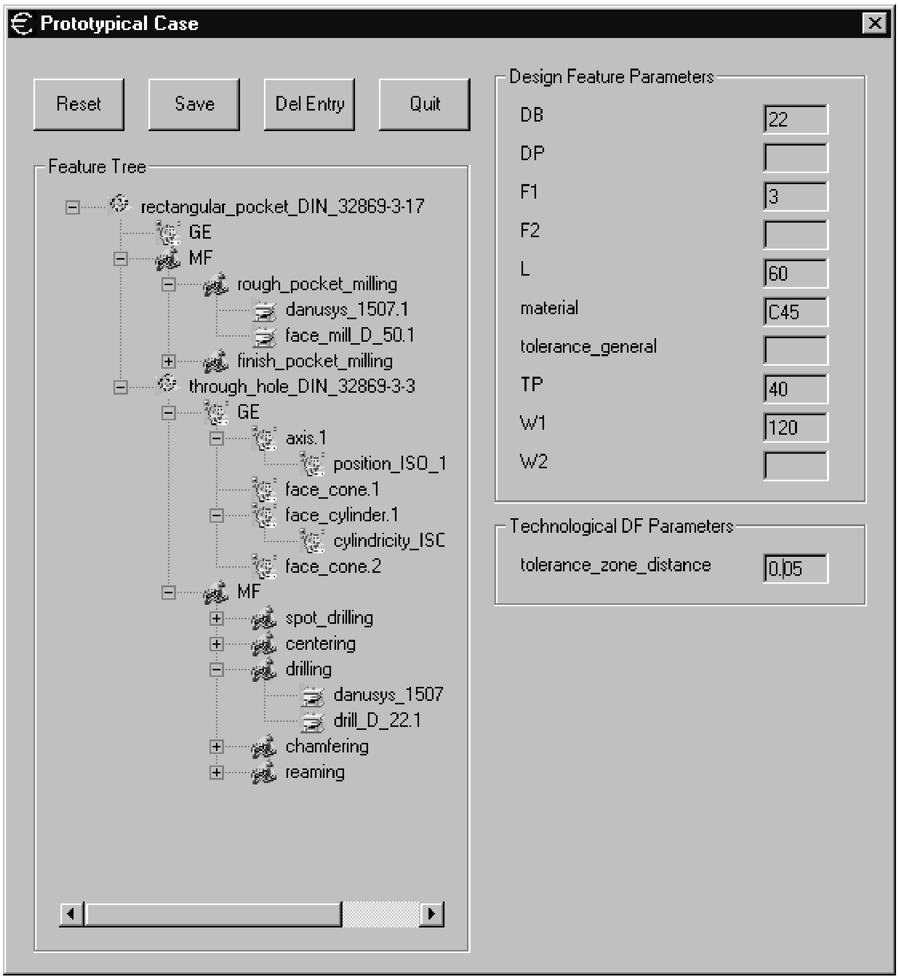

Figure 7.8: Definition of a Prototypical Case.

Finally, a compound of pre-cast rectangular pocket DF, rectangular pocket DF and through hole DF is modeled in the CAD system CATIA V5 and exported. The export file is completed with the technological specification of the form DFs of the CAD model.

For this compound the most similar compound in the case base is retrieved. The most similar case for the compound modeled in the CAD system is the prototypically defined case (see figure 7.8) in the case base. Its solution is adapted to the current compound. Based on this adaptation the resource consumption is newly computed and monetarily valued.

This example has shown how PPR classes are defined, how resource instances are generated, and how the knowledge about manufacturing similarity is represented and prototypical cases defined.

The initial feedback in this exploratory study was that for metal-cutting manufacturing methods this method is more adequate to represent a similarity measure and prototypical cases than if-then rules. Both the allocation function and the similarity matrix were judged to be comprehensible and to support an intuitive definition as well as an exact reflection of the manufacturing aspects. However, it was deemed advisable to reduce the effort needed for similarity definition. An automatic determination of the similarity measure is desired.

At present, the application of CABACO is limited because feature technology in general and feature linking in particular are still in the pipeline. Thus, episodic manufacturing knowledge is currently not at hand. Since the quality of the solution generated increases with the number of cases in the case base and only a very limited number of prototypical cases is currently represented, a thorough validation of the quality of cost estimates is not possible. The next steps to be done are to increase the functionality of CABACO, i.e. representation and processing of generalized manufacturing, inheritance of similarity measures, automatic determination of weighting coefficients, allocation functions, and similarity matrices. Finally, to accompany the development of feature technology and feature linking in order to prepare CABACO to integrated PPR models, i.e. episodic manufacturing knowledge.

7.2 Process Reengineering in the PCP

The introduction of new IT systems in a productive environment is a complex task. The intention of this chapter is to portray the reengineering of processes which are the consequence of the introduction of a system for design-concurrent cost estimation. Although the objective of this work has been to support designers in cost estimation during detail design, the point of view of this chapter is broader. In addition, it takes a look at possible support for other people in the PCP who could benefit from a cost estimation application since the concept does not exclude subsequent modifications of the proposed solution.

Cost estimation applications do not significantly change the workflow of the design engineer. Rather the task of the designer is made easier, since consistent cost information is always at hand. Design engineers can base their design decisions on the information available, and cost-related, tedious iteration loops become a thing of the past. The design engineer is now able to carry out cost-based comparisons between alternative product designs based on a detailed quantification of the resource consumption. In particular, there are no changes needed when the cost estimation application is integrated in a CAD system, e.g. as a cost estimation workbench that forwards the cost information to the part design workbench where it is displayed. The introduction of integrated PPR models as the backbone of cost estimation may from time to time require that new DF types must be added, which might be the responsibility of design engineers. But, this is a task not primarily driven by cost estimation: rather, the application of feature-based product and process modeling calls for this. Similarly cost estimation can support the tasks of product cost managers and module engineers, responsible for assuring that the cost targets are met and that assemblies and sub-assemblies function properly.

Cost estimation applications also expand the scope of activities of the manufacturing engineer and of Procurement. The manufacturing engineer is responsible for delivering the manufacturing knowledge to the knowledge base of the application, i.e. by defining manufacturing similarity (or at least validating automatically generated manufacturing similarity), inputting prototypical cases, including new MFs, and maintaining the resource library. The two latter are tasks the manufacturing engineer might also be burdened with independent of whether a cost estimation system is introduced or not. In addition, the manufacturing engineer has to make sure that the manufacturing knowledge represented in the case base is up-to-date and must, if necessary, maintain the knowledge. The procurement agent is, in addition, occupied with maintaining the database by ensuring the availability of the prices for company-external activities, i.e. world market prices and prices offered for raw material, prices paid and offered for rough parts and services, and by determining a correction coefficient for internal vs. external and rates of price increases. However, this is not done in vain and solely for the benefit of the design engineer, rather it is something that procurement can also benefit from. The controller in turn is not burdened with additional work since such an application relies on data in an ERP system in order to avoid redundant and inconsistent data. For detailed bid preparation, of course, a cost estimation application offers sound advantages since bids can be prepared faster and with less effort.

As cost estimation applies data and knowledge that underlie ongoing changes, it has to be warranted that the quality of the cost estimates is sufficiently high. Solving this problem is the task of a cost model engineer, whose job description has thus far not been fully specified. Quality assurance has to be performed from time to time. Information from post-calculation and production data acquisition is applied here. For the future, integrating automatic assurance functionality in the cost estimation application would be a useful add-on.

Since the background of engineers is mainly technical in nature, further education and qualification is necessary in order to strengthen cost awareness and point out the responsibility toward costs. The objective must target developing the attitude that the cost estimation system is a useful tool in supporting engineers in their daily activities and not to be regarded as superfluous. If the cost estimation system lacks acceptance due to user bias, the potentials of such a system cannot be exploited. Therefore, preparatory work is a must.

It is anticipated that, despite the incidental work for representing and maintaining manufacturing knowledge and for assuring the quality of the cost estimates, product development costs can be reduced because of fewer expenses overall. But, with regard to the responsibility for product costs, the company should take this chance and, instead of cutting the nascent capacities, apply them to generate more design alternatives to be able to choose the one with the lowest costs. It is presupposed that the costs that can thus be saved will exceed the costs that can be saved in product development.

7.3 Summary

In chapter 2, an overview on the product creation environment has been given. The product creation process has been described and feature technology has been introduced. Subsequently, the costing and product cost management have been explained. Chapter 2.4 has given a survey on cost estimation, an introduction to the cost paradox and a description and classification of the different methodological approaches for cost estimation in the scientific context. Furthermore, it has provided an overview of recent work in the field of cost estimation and has described typical methods deployed. Chapter 2.5 has elaborated on process planning; in particular it has focused on traditional process planning and on how process planning is supported by computers. Chapter 2.6 has concluded with the shortcomings and potentials of concepts and tools for cost estimation and process planning with regard to the objectives of this thesis. In chapter 3 knowledge processing in general and the methodology of CBR in particular have been introduced. Because of the importance for the retrieval step in CBR, the main characteristics of similarity have been described. Chapter 4 has described the development of the rough concept for cost estimation with a focus on detail design. The requirements have been refined and the objective to estimate the production costs has been decomposed into the provision of prices to be paid for company-external activities, the estimation of direct manufacturing costs and the computation of overhead costs. Chapter 5 has focused on the cost estimation model for the direct manufacturing costs. For this reason, partial models – product, process, and resource – and their integration have been described. Chapter 5.5 has also discussed how to represent generalized manufacturing knowledge required for cost estimation in integrated PPR models and has therefore introduced generative integration relations. In contrast to this, chapter 5.6 focused on the reuse of specific manufacturing knowledge in data sets and in prototypical cases. Of special interest were the retrieval and the reuse steps of the methodology of CBR with regard the cost estimation in detail design. Chapter 5.7 has dealt with manufacturing similarity, in particular with manual definition and automatic determination of manufacturing similarity. The variance has been computed for the automatically determined manufacturing similarity in order to indicate the certainty related to the correctness of each of them. Chapter 5.8 has brought together the parts of the cost estimation model for direct manufacturing costs. The prototypical implementation of CABACO has been described in chapter 6 and the illustration of CABACO in the metal-cutting domain in chapter 7.

Chapter 8

Concluding Remarks

8.1 Conclusion

Enterprises today are faced with the necessity to cut product costs by developing products which will incur lower production costs, in particular, and, furthermore, by designing desirable products in an ever-dwindling time span. The achievable profit is usually related to the product costs. Hence, it is insufficient to do accounting in retrospect: instead there is a need to proactively analyze and continuously control product costs. However, there is neither an appropriate tool nor an adequate methodology for design-concurrent cost estimation. Thus, since the design engineer is only insufficiently supported in developing cost-optimal products, the objective of this thesis has been to develop a concept for concurrent estimation of production costs that contributes to the design of economically viable products and accelerates the product development process. This chapter now intends to conclude on the results of the research described in this thesis.

This paragraph will disclose the scientific procedure underlying this research work. The requirements for a cost estimation concept have been derived from the current situation as described above and the desire to change it. They have been further refined based on the results of the state-of-the-art literature review. Thus, an issue of practical interest has been captured and elaborated. To avoid a rash decision-making that would have excluded promising solutions or proper technologies, all areas that are in any way related to cost estimation have been examined, of course including the state of the art in this field. Since only generative-analytical models provide a detailed and differentiated cost structure that enables possibilities for cutting product costs to be revealed and since they are based on sequences of manufacturing operations and their monetary evaluation, the state of the art in process planning has also been closely examined. The PCP has been looked at with respect to current and future trends. As generating sequences of manufacturing operations relies on manufacturing knowledge, knowledge processing was also of primary interest. Thus, technologies that may be of interest here have been identified and evaluated on the basis of the requirements to be met. Subsequently, three partial models, one for each aspect, have been developed: together they shape the model for the estimation of direct manufacturing cost in detail design. The cost estimation partial model for direct manufacturing costs has then been further detailed since direct manufacturing costs are the lever used by the design engineer to cut production costs and contain fewer soft facts than prices to be paid for external activities. A portion of this cost estimation model for direct manufacturing costs has been implemented in CABACO with initial feedback gained in an exploratory study. The results of this research work have been applied for advisement in practice, which bends the bow from the challenges in practice to a concept for cost estimation. Of course, the complete concept needs to be implemented and CABACO further applied.

The concepts and tools for cost estimation that have been described show considerable shortcomings with respect to the objective of this thesis. The integration of cost estimation in the product development process and the possibility of design-concurrent usage are not solved satisfactorily. In contrast to generative-analytical models, statistical models and analogy models determine costs in a lump-sum fashion in the vast majority of cases. They are, in particular, not able to identify the cost-driving product characteristics of the products to be calculated because of an insufficient degree of detail and differentiation and, thus, do not permit cost-based comparisons between alternative products.

The absence of an explicit domain model which largely holds true for manufacturing in mechanical engineering, the difficulty in acquiring the knowledge necessary to generate the domain model, and the effort involved in keeping it up to date have been insufficiently considered in the past and remain to a great extent unsolved. This applies for both the generative-analytical concepts and tools as well as for the concepts and tools for automatic process planning. The CBR concepts elaborated for planning and cost estimation are still far away from applicability. In particular, the interpretation of the product model including the interdependencies and the retrieve and reuse steps have not been elucidated sufficiently. How the similarity measure is to be defined and how similarity assessment is to be carried out have, for the most part, been neglected. To conclude, none of the methods currently employed for the design-concurrent cost estimation of production costs is as effective and efficient as required.

In the absence of an explicit and complete domain model, the most promising approach has been to apply and extend the methodology of CBR to the needs of cost estimation together with the possibility to represent and process generalized manufacturing knowledge. This thesis picks up the inadequacies mentioned above – interpretation of the product model and its inherent interdependencies, retrieve and reuse steps, and definition of similarity – and brings them to proper solutions.

The objects (form DFs, MFs, resources, etc.), either abstract or instantiable, and their parameters and the relations between these objects (either within a single partial model, e.g. interaction relation, or between different partial models, e.g. an *is_machined_as* relation) that are of interest for cost estimation have been identified and represented in the integrated PPR model, which enables the interpretation of the product model, turns data in the data resources into information, and serves as the basis for the refinement of the retrieve and reuse steps.

The objective of the retrieve step has been to find the most similar cases and to evaluate each solution using organizational parameters. The retrieve step has taken into consideration that manufacturing a form DF is contingent not only on its characteristics and its technological DFs but also on adjacent form DFs including their technological DFs. Similarity for these compounds is assessed on the basis of manufacturing aspects. In order that such aspects may be represented, two ways for manufacturing similarity representation have been introduced: allocation functions for numerical parameters and similarity matrices for nominally-scaled and discrete parameters. Both have enabled an intuitive definition and exact reflection of the manufacturing aspects. A manual definition of the similarity measure has been made for comprehensible portions of a compound. Similarity assessment has weighted these individual similarities and applies them to compute the similarity for the compound. In addition to the alternatives for passing similarity measures from parent to child classes, this research has presented further possibilities to reduce the effort for similarity definition. These are automatic determination of weighting coefficients, allocation functions, and similarity matrices based on the data contained in the data resources. The variance has been computed for these items in order to indicate the certainty related to the correctness of each of them.

The reuse step discussed has adapted the retrieved solution to the current geometric parameters of the CAD product model. Thus, resource consumption has been computed utilizing up-to-date information. The operations contained in the pool of operations which originate either from the reuse of specific manufacturing knowledge or the application of generalized manufacturing knowledge have been unified and any additional operations required have been instantiated to this pool. Generative IRs contain the knowledge necessary to distinguish whether or not these additional operations are necessary. Thus the generative IRs have been introduced to represent generalized manufacturing knowledge. To further specify these generative IRs, additional relation types have been introduced. These relations enable manufacturing knowledge to be represented in integrated PPR models. In each case, the manufacturing knowledge (generalized or specific domain knowledge) is separated from the processing knowledge, i.e. manufacturing knowledge can be added, deleted or modified at run-time.

The refined requirements as described in chapter 4.1 have largely been fulfilled. An evaluation of the level of accuracy – as also required in chapter 4.1 – has been left open. Since domain knowledge is separated from the processing knowledge, the level of accuracy is contingent not only on the processing knowledge but also on the quality of the domain knowledge. Furthermore, if only a few cases are represented in the case base, the quality of the system is expected to be poor, but if the number of cases grows, the quality is expected to increase commensurately. Thus, the level of accuracy has to be evaluated in the real product creation environment. The model also seems to be applicable outside the model space it has been designed for. In addition to the application as an informational and decision-support system for capacity and production planning, cycle-time harmonization, machine facility design etc., it is applicable as an assisting system for process planning, NC machining programming, etc. Thus, the general validity of the model is not restricted.

The ease of acquisition and maintenance of manufacturing knowledge emphasize the practical value of this thesis. The model for cost estimation in detail design contributes to the design of production cost-minimum products and accelerates the product development process with a decrease in product development costs. In conclusion, the objective and the desired benefits have been met.

8.2 Outlook

Future work must evaluate the cost estimation model in empirical studies in the real manufacturing environment. Particularly interesting are the representation of generalized manufacturing knowledge, the reuse of specific manufacturing knowledge, and finally the quality of the cost estimate differentiating between the prices to be paid and direct manufacturing costs. Therefore, the functionality of CABACO needs to be extended. This, of course, necessitates feature-based design and the accessibility of data sets with manufacturing knowledge in integrated PPR models the absence of which currently prevents the prototypical system from a broader evaluation. Future work must also be concerned with the most convenient way of presenting the result of the cost estimation to the design engineer.

Future research must be directed towards five key goals:

- Transferability of the cost estimation model to embodiment design.
- Extendibility of the model from the estimation of discrete mechanical parts to assemblies.
- Consideration of information generated in concurrent engineering processes.
- Quality assurance of the cost estimates automatically based on data from post-calculation and production data acquisition.
- Integration of inherent dependencies between the products to be manufactured and the cost rates.

Since the possibility of exerting an impact on costs is greater in embodiment design than in detail design, the idea of future research work in the field of cost estimation should be to determine the production costs expected to be incurred at as early a stage as possible and again to determine and depict these costs as detailed and transparently as possible. This of course has to take into account a continuous applicability of the cost estimation model.

The parts in an assembly are not independent of each other: the modification of one part in the assembly usually necessitates the adaptation of other parts. Since product development is always concerned with a global instead of a local cost minimum, it will be unsatisfactory in the future not to be able to anticipate the impact of a modification of a single part based on the total costs of the assembly.

In concurrent engineering, the intention is to bring product development and job planning more and more in line. Furthermore, the model described in this thesis is applicable not only in a decision support system but also in an assisting system. In such a system, the user modifies proposed solutions and therefore usually generates information about the manufacturing process of the current part of higher quality than the system does. Thus, the question is how this kind of knowledge can be applied for subsequent cost estimations in its development process, parts of which will undoubtedly have a higher quality than the knowledge in the knowledge base.

Even though the cost estimation model is currently limited to discrete mechanical parts in detail design, this thesis is an important step towards a continuous transparency of costs during the product creation process. It is thus a crucial building block for product cost management and design-for-X. Further research work is necessary in order to obtain cost estimates for assemblies in all phases of the product development process.

References

Aamodt, 1995	Aamodt, A. (1995). Knowledge acquisition and learning by experience – The role of case-specific knowledge. In: *Machine Learning and Knowledge Acquisition: Integrated Approaches*, (G. Tecuci, and Y. Kodratoff (Eds.)), Academic Press, New York, pp. 197-245.
Aamodt and Plaza, 1994	Aamodt, A., and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations and system approaches, *AI Communications*, Vol. 7, No. 1, pp. 39-59.
Aha, 1998	Aha, D.W. (1998). Feature weighting for lazy learning algorithms In: *Feature Extraction, Construction and Selection: A Data Mining Perspective*, (H. Liu, and H. Motoda (Eds.)), Kluwer, Norwell MA.
Asiedu and Gu, 1998	Asiedu, Y., and Gu, P. (1998). Product life cycle cost analysis: state of the art review, *Int. Journal of Production Research*, Vol. 36, No. 4, pp. 883-908.
Asiedu et al., 2000	Asiedu, Y., Besant, R.W., and Gu, P. (2000). Simulation-based cost estimation under economic uncertainty using kernel estimators, *Int. Journal of Production Research*, Vol. 38, No. 9, pp. 2023-2035.
ASME B46.1–2002	Surface texture (surface roughness, waviness and lay).
ASME Y14.5M–1994	Dimensioning and tolerancing.
Balasubramanian and Herrmann, 1997	Balasubramanian, S., and Herrmann, J.W. (1999). *Using neural networks to generate design similarity measures for variant fixture planning*, University of Maryland, Baltimore, USA.
Baier, 2002	Baier, J. (2002). *Ein Beitrag zur simulationsgestützten Kosten-analyse auf Prozesskostenbasis in der Nutzfahrzeugproduktion*, Ph.D. thesis, Shaker, Aachen, Germany. (in German)
Becker, 1990	Becker, J. (1990). Entwurfs- und konstruktionsbegleitende Kostenkalkulation, *Kostenrechnungspraxis*, Vol. 34, No. 6, pp. 353-358. (in German)
Becker and Prischmann, 1994a	Becker, J., and Prischmann, M. (1994). Konstruktionsbegleitende Kalkulation mit Neuronalen Netzen, *Kostenrechnungspraxis*, Vol. 38, No. 3, pp. 167-171. (in German)
Becker and Prischmann, 1994b	Becker, J., and Prischmann, M. (1994). *VESKONN – Prototypische Umsetzung eines modularen Konzepts zur Konstruktionsunterstützung mit konnektionistischen Methoden*, Westfälische Wilhelms-Universität Münster, Germany. (in German)
Belkin and Croft, 1987	Belkin, N., and Croft, W.B. (1987). Retrieval techniques, In: *Annual Review of Information Science and Technology*, (M.E. Williams (Ed.)), Vol. 22, pp. 109-145.
Bergmann, 1996a	Bergmann, R. (1996). Lernende Systeme – Teil 1: Symbolische Methoden, http://wwwagr.informatik.uni-kl.de/~bergmann/LS/ls96_97.html. (in German)
Bergmann, 1996b	Bergmann, R. (1996). *Effizientes Problemlösen durch flexible Wiederverwendung von Fällen auf verschiedenen Abstraktionsebenen*, Ph.D. thesis, infix, Sankt Augustin. (in German)

Bergmann, 1998 — Bergmann, R. (1998). On the use of taxonomies for representing case features and local similarity measures. *Proc. of the 6th German Workshop on Case-Based Reasoning,* (L. Gierl, and M. Lenz (Eds.)), IMIB Series, Vol. 7, pp. 23-32.

Bergmann and Stahl, 1998 — Bergmann, R., and Stahl, A. (1998). Similarity measures for object-oriented case representations. *Proc. of the European Workshop on Case-Based Reasoning*, (B. Smyth, and P. Cunningham (Eds.)), Springer.

Binder, 1998 — Binder, M. (1998). *Produktkostenmanagement in Entwicklung und Konstruktion*, Ph.D. thesis, Gabler, Wiesbaden. (in German)

Blocher et al., 2002 — Blocher, E.J., Chen, K.H., and Lin, T.W. (2002*). Cost management: A strategic emphasis*, McGraw-Hill, Boston.

Bode, 1998 — Bode, J. (1998). Decision support with neural networks in the management of research and development – Concepts and applications to cost estimation, *Information & Management*, Vol. 34, No. 1, pp. 33-40.

Bode, 2000 — Bode, J. (2000). Neural networks for cost estimation: simulations and pilot application, *Int. Journal of Production Research*, Vol. 38, No. 6, pp. 1231-1254.

Chang and Chang, 2000 — Chang, P.-T., and Chang, C.-H. (2000). An integrated artificial intelligent computer-aided process planning system, *Int. Journal of Computer Integrated Manufacturing*, Vol. 13, No. 6, pp. 483-497.

Chase, 1999 — Chase, K.W. (1999). Tolerance allocation methods for designers, *Int. Journal of Integrated Manufacturing*, Vol. 13, No. 6, pp. 483-497.

Craw et al., 2001 — Craw, S., Jarmulak, J., and Rowe, R. (2001). Maintaining retrieval knowledge in a case-based reasoning system, *Computational Intelligence*, Vol. 17, No. 2, pp. 346-363.

Ciesla, 1997 — Ciesla, M. (1997). *Feature-basierte Messplanung für Koordinatenmessmaschinen*, Ph.D. thesis, Techn. Univ. Berlin, Germany. (in German)

CostDesigner, 1998 — CostDesigner 98 demonstration software.

Cunningham, 1998 — Cunningham, P. (1998). CBR: Strenghts and weaknesses. *Proc. of 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, (A.P. del Pobil, J. Mira, and M. Ali (Eds.)), pp. 517-523.

Cunningham and Bonzano, 1999 — Cunningham, P., and Bonzano, A. (1999). Knowledge engineering in a real world case-based reasoning application, *Knowledge Based Systems*, Vol. 12, pp. 372-379.

Debuschewitz, 1999 — Debuschewitz, M. (1999). *Integrierte Methodik und Werkzeuge zur herstellkostenorientierten Produktentwicklung*, Ph.D. thesis, Springer, Berlin.

DIN 13-1–1999 — General purpose ISO metric screw threads – Part 1: Nominal sizes for 1 mm to 68 mm diameter coarse pitch threads.

DIN 199-2–1977 — Begriffe im Zeichnungs- und Stücklistenwesen – Stücklisten. (in German)

DIN 4760–1982 — Form deviations – Concepts and classification system.

DIN 6773–2001 — Wärmebehandlung von Eisenwerkstoffen – Darstellung und Angaben wärmebehandelter Teile in Zeichnungen. (in German)

DIN 7167–1987 Relationship between tolerances of size, form and parallelism.

DIN 8580–1985 Manufacturing methods – Classification.

DIN 32869-2–2002 Technische Produktdokumentation – Dreidimensionale CAD-Modelle – Anforderungen an Attribute. (in German)

DIN 32869-3–2002 Technische Produktdokumentation – Dreidimensionale CAD-Modelle – Funktionselemente. (in German)

DIN 32990-1–1989 Kosteninformationen – Berechnungsgrundlagen – Kalkulationsarten und -verfahren. (in German)

DoD, 1999 Department of Defense (1999). *Parametric estimating handbook*, http://www.jsc.nasa.gov/bu2/PCEHHTML/pceh.htm.

Dubbel and Beitz, 1995 Dubbel, H., and Beitz, W. (1995). *Taschenbuch für den Maschinenbau*, Springer, Berlin. (in German)

Dreyfus and Dreyfus, 1986 Dreyfus, H.L, and Dreyfus, S.E. (1986). *Mind over machine*, Blackwell, Oxford.

Drucker, 1993 Drucker, P.F. (1993). *Post-capitalist society*, Butterworth-Heinemann, Oxford.

Drury, 2000 Drury, C. (2000). *Management and cost accounting*, Thomson Learning, London.

Dürr and Ewert, 1998 Dürr, H., and Ewert, W. (1998). Learning rules for sequence generation in production planning. *Proc. of the CIRP International Seminar on Intelligent Computation in Manufacturing Engineering*, (R. Teti (Ed.)), Capri, Italy, pp. 83-90.

Duverlie and Castelain, 1999 Duverlie, P., and Castelain, J.M. (1999). Cost estimation during design step: parametric method versus case based reasoning method, *Int. Journal of Advanced Manufacturing Technology*, Vol. 15, pp. 895-906.

Eaglesham, 1998 Eaglesham, M.A. (1998). *A decision support system for advanced composites manufacturing cost estimation*, Ph.D. thesis, Virginia State University, Blacksburg, USA.

Endebrock, 2000 Endebrock, K. (2000). *Ein Kosteninformationsmodell für die frühzeitige Kostenbeurteilung in der Produktentwicklung*, Ph.D. thesis, Ruhr-Universität Bochum, Germany. (in German)

Endebrock and Welp, 1999 Endebrock, K., and Welp, E.G. (1999). Object-oriented product and process modeling for transparent cost estimation. *Proc. of ICED99*, pp. 493-496.

Ehrlenspiel et al., 1999 Ehrlenspiel, K., Kiewert, A., and Lindemann, U. (1999). *Kostengünstig Entwickeln und Konstruieren*, Springer, Berlin. (in German)

Eisinger, 1997 Eisinger, B. (1997). *Konstruktionsbegleitende Kalkulation – Modell eines effizienten Kosteninformationssystems*, Ph.D. thesis, Dt. Univ.-Verl., Wiesbaden. (in German)

Eitrich, 1996 Eitrich, O. (1996). *Prozessorientiertes Kostenmodell für die entwicklungsbegleitende Vorkalkulation*, PhD thesis, Univ. Karlsruhe, Germany. (in German)

Eversheim and Michalas, 2001 Eversheim, W., and Michalas, N. (2001). *Integration of process planning and production scheduling: benefits and difficulties*, http://www.3s.hmg.inpg.fr/ci/GAMA/01Eversheim.pdf.

Eversheim et al., 1998 — Eversheim, W., Neuhausen, J., and Sesterhenn, M. (1998). Design-to-cost for production systems, *CIRP Annals – College International de Recherches pour la Production*, Vol. 47, No. 1, pp. 357-360.

Ewert and Wagenhofer, 2000 — Ewert, R., and Wagenhofer, A. (2000). *Interne Unternehmens-rechnung*, Springer, Berlin. (in German)

Ewert et al., 1995 — Ewert, W., Leidholdt, F., and Dürr, H. (1995). *Repräsentation technologischer Wissensbasen für die Wissensakquisition durch Maschinelles Lernen in der generierenden Arbeitsplanerstellung*, Chemnitz University of Technology, Germany. (in German)

Feller, 1992 — Feller, A.H. (1992). *Kalkulation in der Angebotsphase mit dem selbsttätig abgeleiteten Erfahrungswissen der Arbeitsplanung,* PhD thesis, Univ. Karlsruhe, Germany. (in German)

Franz and Kajüter, 1997 — Franz, K.P., and Kajüter, P. (1997). *Kostenmanagement: Wettbewerbsvorteile durch systematische Kostensteuerung*, Schäffer-Poeschel, Stuttgart. (in German)

Frech, 1998 — Frech, J.T. (1998). *Ein Verfahren zur integrierten, prozessbegleitenden Vorkalkulation für die kostengerechte Konstruktion,* Ph.D. thesis, Springer, Berlin. (in German)

Frenzel et al., 2001a — Frenzel, M., Frank, D., Haasis, S., Layer, A., and Schulze, H. (2001). Konstruktionsbegleitende Kostenkalkulation auf Basis der Feature-Technologie, *CAD-CAM Report*, Vol. 20, No. 3, pp. 32-40. (in German)

Frenzel et al., 2001b — Frenzel, M., Frank, D., Haasis, S., Layer, A., and Schulze, H. (2001). Featurebasierte Kostenkalkulation in der Produktentwicklung, *VDI-Z*, Vol. 143, No. 3, pp. 75-77. (in German)

Frenzel et al., 2001c — Frenzel, M., Frank, D., Haasis, S., Layer, A., and Schulze, H. (2001). Lösungsansatz für Feature- und Web-basierte Kostenkalkulation am Beispiel Betriebsmittelbau, *Werkstatts-technik*, Vol. 91, No. 6, pp. 349-351. (in German)

Frost, 1986 — Frost, R.A. (1986). *Introduction to knowledge based systems*, Collins, London.

Geelink, 1996 — Geelink, R. (1996). *Flexible definition of form features*, Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Gerhard, 1994 — Gerhard, E. (1994). *Kostenbewusstes Entwickeln und Konstruieren: Grundlagen und Methoden zur Kostenbestimmung und Kostenabschätzung während eines entwicklungs- und herstellkostenorientierten Vorgehens*, expert, Renningen-Malmsheim. (in German)

Gerken, 2000 — Gerken, H. (2000). *Management von Erfahrungen mit einem Assistenzsystem für die Arbeitsplanung*, PhD thesis, Univ. Karlsruhe, Germany. (in German)

Giebels, 2000 — Giebels, M. (2000). *EtoPlan – a concept for concurrent manufacturing planning and control*, Ph.D thesis, University of Twente, Enschede, The Netherlands.

Göker, 1998 — Göker, M.H. (1998). Ähnlichkeit technischer Objekte und Verfahren zum Ermitteln von Gewichtungsfaktoren, *Konstruktion*, Vol. 50, No. 5, pp. 47-52. (in German)

Goos, 1996 — Goos, K. (1996). *Fallbasiertes Klassifizieren. Methoden, Integration und Evaluation*, Ph.D. thesis, infix, Sankt Augustin. (in German)

Gröner, 1991 — Gröner, L. (1991). *Entwicklungsbegleitende Vorkalkulation*, Ph.D. thesis, Springer, Berlin. (in German)

Grundmann, 1994 — Grundmann, M. (1994). *Entwicklung eines Kalkulationsinstrumentariums für Spritzgiessbetriebe auf der Basis von Ähnlichkeitsbetrachtungen*, Ph.D. thesis, Verlag der Augustinus Buchhandlung, Aachen. (in German)

Haasis, 1995 — Haasis, S. (1995). *Wissens- und featurebasierte Unterstützung der Konstruktion von Stirnradgetrieben unter besonderer Berücksichtigung des Gussgehäuses*, Ph.D. thesis, VDI-Verlag, Düsseldorf. (in German)

Haasis et al., 1999 — Haasis, S., Frank, D., Rommel, B., and Weyrich, M. (1999). Feature-basierte Integration von Produktentwicklung, Prozessgestaltung und Ressourcenplanung. In: *VDI-Berichte 1497*, VDI-Verlag, Düsseldorf, pp. 333-348. (in German)

Haasis et al., 2000a — Haasis, S., Bär, T., Rothenburger, G., Schulze, H., and Ströhle, H. (2000). User-oriented inspection modeling for powertrain engineering. *Proc. of Science Days*, Sindelfingen, Germany, pp. 109-120.

Haasis et al., 2000b — Haasis, S., Ströhle, H., Karthe, T., and Pfeifle, J. (2000). Feature-basierte Prüfmodellierung. In: *VDI-Berichte 1569*, VDI-Verlag, Düsseldorf, pp. 199-212. (in German)

Haasis et al., 2001a — Haasis, S., Layer, A., and Schulze, H. (2001). Feature technology: Added value for the engineering process – Cost calculation and experience transfer. *Proc. of FEATS*, Valenciennes, France.

Haasis et al., 2003 — Haasis, S., Frank, D., Rommel, B., and Weyrich, M. (2003). Feature-based integration of product, process and resource, In: *Feature based product life-cycle modeling*, (G.J. Olling, and R. Soenen (Eds.)), Kluwer, Boston, pp. 93-108.

Häcker and Stapf, 1998 — Häcker, H., and Stapf, K.-H. (1998). *Dorsch Psychologisches Wörterbuch*, Hans Huber, Bern. (in German)

Halevi and Weill, 1995 — Halevi, G., and Weill, R.D. (1995). *Principles of process planning – A logical approach*, Chapman & Hall, London.

Heine, 1995 — Heine, A. (1995). *Entwicklungsbegleitendes Produktkostenmanagement – Gestaltung des Führungssystems am Beispiel der Automobilindustrie*, Ph.D. thesis, Dt. Univ.-Verlag, Wiesbaden. (in German)

Herrmann, 1997 — Herrmann, J. (1997). *Maschinelles Lernen und wissensbasierte Systeme*, Springer, Berlin. (in German)

Hintz, 1996 — Hintz, A. (1996). *Konzeption und Implementierung eines Systems zur automatischen Generierung von Arbeitsplänen unter Berücksichtigung alternativer Fertigungsmethoden*, Ph.D. thesis, Shaker, Aachen. (in German)

Horváth et al., 1997 — Horváth, P., Gleich, R., and Scholl, K. (1997). Vergleichende Betrachtung der bekanntesten Kalkulationsmethoden für das kostengünstige Konstruieren. In: *Frühzeitiges Kostenmanagement*, (W. Männel (Ed.)), Gabler, Wiesbaden, pp. 111-132. (in German)

Ho, 1997 — Ho, C.-C. (1997). *A feature-based process planning and automatic numerical control programming*, Ph.D. thesis, University of Utah, Salt Lake City, USA.

Houten, 1991	Houten, F.J.A.M. van (1991). *PART: a computer aided process planning system*, Ph.D. thesis, University of Twente, Enschede, The Netherlands.
Houten and Kals, 1999	Houten, F.J.A.M. van, and Kals, H.J.J. (1999). Global consistency of tolerances. *Proc. of the 6th CIRP international seminar on computer-aided tolerancing*, (F.J.A.M. van Houten, and H.J.J. Kals (Eds.)), Enschede, The Netherlands.
Humienny, 2001	Humienny, Z. (2001). *Geometrical product specifications: a course for technical universities*, http://www.qfm.uni-erlangen.de/gps/.
ISO 286-1–1988	ISO system of limits and fits – Part 1: Bases of tolerances, deviations and fits.
ISO 1101–1983	Geometrical tolerancing – Tolerancing of form, orientation, location and run-out – Generalities, definitions, symbols, indications on drawings.
ISO 1302–2002	GPS – Indication of surface texture in technical product documentation.
ISO 2692–1988	Technical drawings – Geometrical tolerancing – Maximum material principle.
ISO 2768-1–1989	General tolerances – Part 1: Tolerances for linear and angular dimensions.
ISO 2768-2–1989	General tolerances – Part 2: Geometric tolerances.
ISO 5459–1981	Technical drawings – Dimensional tolerancing – Datums and datum-systems for geometrical tolerances.
ISO 8015–1985	Technical drawings – Fundamental tolerancing principle.
ISO 8062–1994	Casting – System of dimensional tolerances and machining allowances.
ISO 8785–1998	GPS – Surface imperfections – Terms, definitions and parameters.
ISO/TR 14638–1995	Geometrical product specification (GPS) – Masterplan.
Jarmulak et al., 2001	Jarmulak, J., Craw, S., and Rowe, R. (2001). Using case-based data to learn adaptation knowledge for design, *Proc. of the 17th IJCAI*, pp. 1011-1016.
Jorden, 2001	Jorden, W. (2001). *Form- und Lagetoleranzen*, Hanser, München. (in German)
Kals et al., 1999	Kals, H.J.J., Brinke, E. ten, D. Lutters, and Streppel, A.H. (1999). Integrated cost estimation based on information management. In: *Production Engineering – research and development in Germany*, Vol. 6, No. 2, pp. 125-130.
Katzenbach et al., 2001	Katzenbach, A., Haasis, S., and Schulze, H. (2001). Bridging the gap – Features as the backbone for integrated engineering. *Proc. of the Fourth Workshop on Current CAx-Problems*, Tokyo, Japan, pp. 35-44.
Kiritsis et al., 1999	Kiritsis, D., Neuendorf, K-P., and Xirouchakis, P. (1999). Petri net techniques for process planning cost estimation, *Advances in Engineering Software,* Vol. 30, No. 6, pp. 375-387.
Kiritsis and Xirouchakis, 2000	Kiritsis, D., and Xirouchakis, P. (2000). DEVIPLAN: a bid preparation system for mechanical SME. *Proceedings of Design Technical Conferences*, Baltimore, Maryland.

Kolodner, 1993 — Kolodner, J.L. (1993). *Case-based reasoning*, Morgan Kaufmann, San Mateo.

König, 1994 — König, T. (1994). *Konstruktionsbegleitende Kalkulation auf der Basis von Ähnlichkeitsvergleichen*, Ph.D. thesis, Josef Eul, Bergisch Gladbach. (in German)

Krause, 1997 — Krause, F.-L. (1997). Qualitäts- und Kostenmanagement beginnen in der Produktentwicklung, *Konstruktion*, Vol. 49, No. 7-8, p. 3. (in German)

Kümper, 1996 — Kümper, R. (1996). *Ein Kostenmodell zur verursachungsgerechten Vorkalkulation*, Ph.D. thesis, Shaker, Aachen. (in German)

Langley, 1996 — Langley, P. (1996). *Elements of machine learning – An overview of machine learning*, Morgan Kaufmann.

Layer et al., 2001 — Layer, A., Haasis, S., and Houten, F.J.A.M. van (2001). Feature-based, design-concurrent cost calculation using case-based reasoning (CBR). *Proc. of DETC*, Pittsburgh, Pennsylvania.

Layer et al., 2002a — Layer, A., Brinke, E. ten, Houten, F.J.A.M. van, Kals, H.J.J., and Haasis, S. (2002). Recent and future trends in cost estimation, *Int. Journal of Computer Integrated Manufacturing*, Vol. 15, No. 6, pp. 499-510.

Layer et al., 2002b — Layer, A., Haasis, S., and Houten, F.J.A.M. van (2002). Reuse of episodic manufacturing knowledge for design-concurrent cost calculation. *Proc. of the 3rd CIRP International Seminar on Intelligent Computation in Manufacturing Engineering*, (R. Teti (Ed.)), Ischia, Italy, pp. 157-161.

Leake, 1996 — Leake, D. (1996). CBR in context: The present and the future. In: *Case-Based Reasoning: Experiences, Lessons & Future Directions*, (D. Leake (Ed.)), AAAI-Press.

Leber, 1995 — Leber, M. (1995). *Entwicklung einer Methode zur restriktionsgerechten Produktgestaltung auf der Basis von Ressourcenverbräuchen*, Ph.D. thesis, Shaker, Aachen. (in German)

Leibl, 1998 — Leibl, P. (1998). *Entwicklung eines featureorientierten Hilfsmittels für die Konstruktion kostengünstiger Produkte – Eine exemplarische Untersuchung an Blechteilen*, Ph.D. thesis, TU Ilmenau, Germany. (in German)

Leibl et al., 1999 — Leibl, P., Hundal, M., and Hoehne, G. (1999). Cost calculation with a feature-based CAD system using modules for calculation, comparison and forecast, *Journal of Engineering Design*, Vol. 10, No. 1, pp. 93-102.

Leidich et al., 2001 — Leidich, E., Jurklies, I., and Schumann, F.J. (2001). Fuzzybasierte Kostenprognose in der Konzeptphase, *Konstruktion*, Vol. 53, No. 3, pp. 83-87. (in German)

Lenz et al., 1996 — Lenz, M., Burkhard, H.-D., Pirk, P., Auriol, E., and Manago, M. (1996). CBR for diagnosis and decision support, *AI Communications*, Vol. 9, No. 3, pp. 138-146.

Levesqueand Lakemeyer, 2000 — Levesque, H.J., and Lakemeyer, G. (2000). *The logic of knowledge bases*, MIT Press, Cambridge.

Liebers, 1998 — Liebers, A. (1998). *An architecture for cost control in manufacturing – The use of cost information in order-related decisions*, Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Lindemann, 1998 — Lindemann, U. (1998). Kosten, Zeit, Qualität, Flexibilität ... "entstehen" in der Konstruktion, *Konstruktion*, Vol. 50, No. 11/12, p. 3. (in German)

Locascio, 2000 — Locascio, A. (2000). Manufacturing cost modeling for product design, *Int. Journal of Flexible Manufacturing Systems*, Vol. 12, No. 2, pp. 207-217.

Lutters, 2001 — Lutters, D. (2001). *Manufacturing integration based on information management*, Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Mannewitz, 1994 — Mannewitz, F. (1994). *Prozessfähige Tolerierung von Bauteilen und Baugruppen – ein Lösungsansatz zur Optimierung der Werkstattfertigung im Informationsverbund zwischen CAD und CAQ*, Ph.D. thesis, Univ.-Gesamthochschule Kassel, Germany. (in German)

Mántaras and Plaza, 1997 — Mántaras, R.L. de, and Plaza, E. (1997). Case-based reasoning: an overview, *AI Communications*, Vol. 10, No. 1, pp. 21-29.

Marefat and Britanik, 1997 — Marefat, M., and Britanik, J. (1997). Case-based process planning using an object-oriented model representation, *Robotics & Computer-integrated manufacturing*, Vol. 13, No. 3, pp. 229-251.

Márkus et al., 1997 — Márkus, A., Váncza, J., and Horváth, M. (1997). Process planning by retrieval and adaptation, *Computers in Industry*, Vol. 33, No. 1, pp. 47-60.

Marri et al., 1998 — Marri, H.B., Gunasekaran, A. and Grieve, R.J. (1998). Computer-aided process planning: A state of the art*, Int. Journal of Advanced Manufacturing Technology*, Vol. 14, No. 4, pp. 261-268.

Mathieu et al., 1998 — Mathieu, L., Clement, A., and Bourdet, P. (1998). Modeling, representation and processing of tolerances, tolerance inspection: a survey of current hypothesis. In: *Geometric design tolerancing: theories, standards and applications*, (H.A. Elmaraghy (Ed.)), Chapman & Hall, London, pp. 1-33.

Mbang et al., 2002a — Mbang, S., Ströhle, H., and Haasis, S. (2002). Intelligent feature model for flexible manufacturing. *Proc. of FAIM*, Dresden, Germany, pp. 116-126.

Mbang et al., 2002b — Mbang, S., Ströhle, H., and Haasis, S. (2002). Trends bei der Feature-basierten Karosserieentwicklung, *CAD-CAM Report*, Vol. 21, No. 11, pp. 50-54. (in German)

Mbang et al., 2003 — Mbang, S., Ströhle, H., Haasis, S., and Rehner, H.-P. (2003). Feature-basierende Anwendungen in der Prozesskette Karosserie, *CAD-CAM Report*, Vol. 22, No. 1, pp. 38-42. (in German)

McLeod and Hanks, 1985 — Collins Concise Dictionary of the English Language (1985). (W.T. McLeod, and P. Hanks (Eds.)), Guild Publishing, London.

Meyer, 1998 — Meyer, S. (1998). *Fuzzy-fallbasierte Kostenschätzung auf der Grundlage eines PDM-Systems*, Chemnitz University of Technology, Germany. (in German)

Meyer, 2000 — Meyer, S. (2000). *Case-based cost estimation at initial stages of design*, Chemnitz University of Technology, Germany.

Meyer, 2001 — Meyer, S. (2001). *Verarbeitung unscharfer Informationen für die fallbasierte Kostenschätzung im Angebotsengineering*, Ph.D. thesis, Chemnitz University of Technology, Germany. (in German)

Minsky, 1986 — Minsky, M. (1986). *The society of mind*, Simon and Schuster, New York.

Morik et al., 1993 — Morik, K., Wrobel, S., Kietz, J.U., and Emde, W. (1993). *Knowledge acquisition and machine learning: Theory, Methods and Applications*, Academic Press, London.

Muñoz-Avila and Weberskirch, 1996 — Muñoz-Avila, H., and Weberskirch, F. (1996). Planning for manufacturing workpieces by storing, indexing and replaying planning decisions. *Proc. of 3rd International Conference on AI Planning Systems*, AAAI-Press.

Neff et al., 2000 — Neff, T., Kokes, M., Mathes, H.D., Hertel, G., and Virt, W. (2000). Front Load Costing – Produktkostenmanagement auf Basis unvollkommener Information, *Kostenrechnungspraxis,* Vol. 44, No. 1, pp. 15-24. (in German)

Neff, 2002 — Neff, T. (2002). *Front Load Costing: Produktkostenmanagement auf der Basis unvollkommener Information*, Ph.D. thesis, Dt. Univ.-Verl., Wiesbaden. (in German)

Nonaka and Takeuchi, 1995 — Nonaka, I., and Takeuchi, H. (1995). *The knowledge-creating company*, Oxford University Press, New York.

Nusswald, 1998 — Nusswald, M. (1998). *Fertigung von Produkten mit Maßketten – Optimierung von Kosten und Durchlaufzeiten*, Ph.D. thesis, Univ. Dortmund, Germany. (in German)

Olfert, 1999 — Olfert, K. (1999). *Kostenrechnung*, Kiehl, Ludwigshafen. (in German)

Ou-Yang and Lin, 1997 — Ou-Yang, C., and Lin, T.S. (1997). Developing an integrated framework for feature-based early manufacturing cost estimation, *Advanced Manufacturing Technology*, Vol. 13, No. 9, pp. 618-627.

Pacyna et al., 1982 — Pacyna, H., Hillebrand, A., and Rutz, A. (1982). Kostenfrüherkennung für Gussteile. In: *Konstrukteure senken Herstellkosten: Methoden und Hilfen*, VDI-Verlag, Düsseldorf, pp. 103-114.

Pahl and Beitz, 1996 — Pahl, G., and Beitz, W. (1996). *Engineering Design: A systematic approach*, Springer, London.

Paulokat and Wess, 1994 — Paulokat, J., and Wess, S. (1994). Planning for machining workpieces with a partial-order, nonlinear planner. *Proc. of AAAI Fall Symposium "Planning and Learning: On to Real Applications"*, AAAI-Press.

Petersohn, 1997 — Petersohn, H. (1997). *Vergleich von multivariaten statistischen Analyseverfahren und künstlichen neuronalen Netzen zur Klassifikation bei Entscheidungsproblemen in der Wirtschaft*, Ph.D. thesis, Peter Lang, Frankfurt/Main. (in German)

Pickel, 1989 — Pickel, H. (1989). *Kostenmodelle als Hilfsmittel zum kostengünstigen Konstruieren*, Ph.D. thesis, Hanser, München. (in German)

Pivert and Rivière, 1999 — Pivert, P. le, and Rivière, A. (1999). Towards realistic simulation of machining processes. *Proc. of the 6th CIRP international seminar on computer-aided tolerancing*, (F.J.A.M. van Houten, and H.J.J. Kals (Eds.)), Enschede, The Netherlands, pp. 55-62.

Polanyi, 1966 — Polanyi, M. (1966). *The tacit dimension*, Doubleday, Garden City.

Puppe, 1991 — Puppe, F. (1991). *Einführung in Expertensysteme*, Springer, Berlin. (in German)

Puppe et al., 2000 — Puppe, F., Stoyan, H., and Studer, R. (2000). Knowledge engineering. In: *Handbuch der Künstlichen Intelligenz*, (G. Görz, C.-R. Rollinger, and J. Schneeberger (Eds.)), Oldenbourg, München, pp. 599-641. (in German)

Rehman, 2000 — Rehman, S. (2000). *Knowledge-based cost modelling for innovative design*, Ph.D. thesis, Cranfield University, Great Britain.

Rehman and Guenov, 1998a — Rehman, S., and Guenov, M.D. (1998). A methodology for modelling manufacturing costs at conceptual design, *Computers and Industrial Engineering,* Vol. 35, No. 3-4, pp. 623-626.

Rehman and Guenov, 1998b — Rehman, S., and Guenov, M.D. (1998). An artificial intelligence approach to innovative design to cost estimation, *Aerogram*, Vol. 9, No. 2, pp. 33-37.

Reischl, 2001 — Reischl, C. (2001). *Simulation von Produktkosten in der Entwicklungsphase*, Ph.D. thesis, TU München, Germany. (in German)

Richter, 1995a — Richter, M.M. (1995). The knowledge contained in similarity measures, Invited Talk on the ICCBR-95, http://wwwagr.informatik.uni-kl.de/~lsa/CBR/Richtericcbr95remarks.html

Richter, 1998 — Richter, M.M. (1998). Introduction. In: *Case-based reasoning: from foundations to applications*, (M. Lenz, H.-D. Burkhard, B. Bartsch-Spörl, and S. Wess (Eds.)), Springer, Berlin, pp. 1-15.

Richter and Althoff, 1999 — Richter, M.M., and Althoff, K.-D. (1999). Similarity and utility in non-numerical domains. In: *Methoden der Wirtschaftswissenschaften*, (W. Gaul, and M. Schader (Eds.)), Physika-Verlag, pp. 403-413.

Richter, 2000 — Richter, M.M. (2000). Fallbasiertes Schliessen. In: *Handbuch der Künstlichen Intelligenz,* (G. Görz, C.-R. Rollinger, and J. Schneeberger (Eds.)), Oldenbourg, pp. 407-430. (in German)

Riesbeck and Schank, 1989 — Riesbeck, C.K., and Schank, R.C. (1989*). Inside case-based reasoning*, Lawrence Erlbaum, Hillsdale, New Jersey.

Salomons, 1995 — Salomons, O.W. (1995). *Computer support in the design of mechanical products*, Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Salomons et al., 1998 — Salomons, O.W., Houten, F.J.A.M. van, and Kals, H.J.J. (1998). Current status of CAT systems. In: *Geometric design tolerancing: theories, standards and applications*, (H.A. Elmaraghy (Ed.)), Chapman & Hall, London, pp. 438-452.

Schaal, 1992 — Schaal, S. (1992). *Integrierte Wissensverarbeitung mit CAD am Beispiel der konstruktionsbegleitenden Kalkulation*, Ph.D. thesis, Hanser, München. (in German)

Schank, 1982 — Schank, R.C. (1982). *Dynamic memory: a theory of reminding and learning in computers and people*, Cambridge Univ. Press, Cambridge.

Scholl, 1997 — Scholl, K. (1997). *Konstruktionsbegleitende Kalkulation – Computergestützte Anwendung von Prozesskostenrechnung und Kostentableaus*, Ph.D. thesis, Vahlen, München. (in German)

Schreiber et al., 1993 — Schreiber, G., Wielinga, B., and Breuker, J. (1993). KADS – *A principled approach to knowledge-based system development*, Academic Press, London.

Schreiber et al., 2001 — Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R. de, Shadbolt, N., Velde, W. van de, and Wielinga, B. (2001). *Knowledge engineering and management – The CommonKADS methodology*, MIT Press, Cambridge.

Schreve et al., 1999 — Schreve, K., Schuster, H.R., and Basson, A.H. (1999). Manufacturing cost estimation during design of fabricated parts. *Proc. of the Institution of Mechanical Engineers*, Vol. 213, No. 7, pp. 731-735.

Seidenschwarz, 1991 — Seidenschwarz, W. (1991). *Target costing*, Univ. of Stuttgart, Germany. (in German)

Shah and Mäntylä, 1995 — Shah, J., and Mäntylä, M. (1995). *Parametric and feature-based CAD/CAM*, Wiley, New York.

Smith, 1996 — Smith, P. (1996). *An introduction to knowledge engineering*, Thomson, London.

Sohlenius, 1992 — Sohlenius, G. (1992). Concurrent engineering, In: *Annals of the CIRP*, Vol. 41, No. 2, pp. 645-655.

Sormaz and Khoshnevis, 1997 — Sormaz, D.N., and Khoshnevis, B. (1997). Process planning knowledge representation using an object-oriented data model, *Int. Journal of Computer Integrated Manufacturing*, Vol. 10, No. 1-4, pp. 92-104.

Sriram, 1997 — Sriram, R.D. (1997*). Intelligent systems for engineering*, Springer, London.

Stachowiak, 1973 — Stachowiak, H. (1973). *Allgemeine Modelltheorie*, Springer, Wien. (in German)

Stark, 1994 — Stark, R. (1994). *Entwicklung eines mathematischen Toleranzmodells zur Integration in (3D-) CAD-Systeme*, Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany. (in German)

Stefik, 1995 — Stefik, M. (1995). *Introduction to knowledge systems*, Morgan Kaufmann, San Francisco.

Steiner, 1996 — Steiner, J.M. (1996). *Rechnergestütztes Kostensenken im praktischen Einsatz*, Ph.D. thesis, Shaker, Aachen. (in German)

Stöckert et al., 1999 — Stöckert, B. Meyer, S., and Rösler, M. (1999). Kostenvorhersagen für das Angebotsengineering komplexer Produkte, *Kostenrechnungspraxis*, Vol. 43, No. 1, pp. 22-29. (in German)

Suh, 1990 — Suh, N.P. (1990). *The principles of design*, Oxford University Press, New York.

Ten Brinke, 2002 — Brinke, E. ten (2002). *Costing support and cost control in manufacturing – A cost estimation tool applied in the sheet metal domain*, Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Thome, 2001 — Thome, O. (2001). *Durchgängige Verarbeitung und Erfassung von Toleranzinformationen*, Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany. (in German)

Thompson, 1999 — Thompson, F. (1999). Cost measurement and analysis. In: *Handbook of Government Budgeting*, (R.T. Meyers (Ed.)), Jossey-Bass, San Francisco.

Tiwari et al., 2001 — Tiwari, M.K., Rama Kotaiah, K., and Bhatnagar, S. (2001). A case-based computer-aided process-planning system for machining prismatic components, *Int. Journal of Advanced Manufacturing Technology*, Vol. 17, No. 6, pp. 400-411.

Trender, 2000 — Trender, L. (2000). *Entwicklungsintegrierte Kalkulation von Produktlebenszykluskosten auf Basis der ressourcenorientierten Prozesskostenrechnung*, Ph.D. thesis, Univ. Karlsruhe, Germany. (in German)

Tsai and Wang, 1999 — Tsai, J.-C., and Wang, W.-W. (1999). Development of a computer-aided tolerancing system in CAD environment. *Proc. of the 6th CIRP international seminar on computer-aided tolerancing*, (F.J.A.M. van Houten, and H.J.J. Kals (Eds.)), Enschede, The Netherlands, pp. 47-54.

Tversky, 1977 — Tversky, A. (1977). Features of similarity, *Psychological Review*, Vol. 84, No. 4, pp. 327-352.

Váncza, 1998 — Váncza, J. (1998). Planning in manufacturing domains controlled by case-based reasoning. In: *Case-Based Reasoning Integrations*, (D. Aha, and J. Daniels (Eds.)), AAAI Press, pp. 161-165.

VDI 2218–1999 — Feature-Technologie. (in German)

VDI 2220–1980 — Produktplanung – Ablauf, Begriffe, Organisation. (in German)

VDI 2221–1993 — Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. (in German)

VDI 2222-1–1997 — Konstruktionsmethodik – Methodisches Entwickeln von Lösungsprinzipien. (in German)

VDI 2225-1–1997 — Konstruktionsmethodik – Technisch-wirtschaftliche Entscheidungen beim Konstruieren – Vereinfachte Kostenermittlung. (in German)

VDI 2234–1990 — Wirtschaftliche Grundlagen für den Konstrukteur. (in German)

VDI 2235–1987 — Wirtschaftliche Entscheidungen beim Konstruieren – Methoden und Hilfen. (in German)

VDI 2800–2000 — Wertanalyse. (in German)

VDI 3633-7-2001 — Simulation von Logistik-, Materialfluss- und Produktionssystemen – Kostensimulation. (in German)

Vin, 1994 — Vin, L.J. de (1994). *Computer aided process planning for the bending of sheet metal components*, Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Wartzack, 2001 — Wartzack, S. (2001). *Predictive Engineering: Assistenzsystem zur multikriteriellen Analyse alternativer Produktkonzepte*, Ph.D. thesis, Universität Erlangen-Nürnberg, Germany. (in German)

Wartzack and Meerkamm, 2000 — Wartzack, S., and Meerkamm, H. (2000). Durchgängige Rechnerunterstützung in der Produktentwicklung durch den Einsatz von semantisch hochwertigen Features, *Konstruktion*, Vol. 52, No. 3, pp. 24-26. (in German)

Watson, 1998 — Watson, I. (1998). CBR is a methodology not a technology. In: *Research & Development in Expert Systems XV*, (R. Miles, M. Moulton, and M. Bramer (Eds.)), pp. 213-223.

Watson and Marir, 1994 — Watson, I., and Marir, F. (1994). Case-based reasoning: a review, *The Knowledge Engineering Review*, Vol. 9, No. 4, pp. 327-354.

Wess, 1996 — Wess, S. (1996). *Fallbasiertes Problemlösen in wissensbasierten Systemen zur Entscheidungs-unterstützung und Diagnostik.*

Grundlagen, Systeme und Entscheidungen, Ph.D. thesis, infix, Sankt Augustin. (in German)

Westkämper, 1997 — Westkämper, E. (1997). Lernfähige Produktion. In: *Stuttgarter Impulse durch Technik und Organisation*, Springer, Berlin, pp. 226-243. (in German)

Wettschereck and Aha, 1995 — Wettschereck, D., and Aha, D.W. (1995). Weighting features. In: *Case-Based Reasoning Research and Development*, (M.M. Veloso, and A. Aamodt (Eds.)), pp. 347-358.

Wilke and Bergmann, 1998 — Wilke, W., and Bergmann, R. (1998). Techniques and knowledge used for adaptation during case-based problem solving. In: *Tasks and methods applied in artificial intelligence*, (A.P. del Pobil, J. Mira, and M. Ali (Eds.)), pp. 497-506.

Wilke et al., 1996 — Wilke, W., Vollrath, I., Althoff, K.-D., and Bergmann, R. (1996). A framework for learning adaptation knowledge based on knowledge light approaches, *Adaptation in CBR: A Workshop at ECAI 1996,* http://citeseer.nj.nec.com/wilke96framework.html.

Wilson and Martinez, 1997 — Wilson, D.R., and Martinez, T.R. (1997). Improved heterogeneous distance functions*, Journal of Artificial Intelligence Research*, Vol. 6, No. 1, pp. 1-34.

Wolfram, 1994 — Wolfram, M. (1994). *Feature-basiertes Konstruieren und Kalkulieren*, Ph.D. thesis, Hanser, München. (in German)

Yang and Lu, 1994 — Yang, H., and Lu, W.F. (1994). Case adaptation in a case-based process planning system. *Proc. of the Second International Conference on Artificial Intelligence Planning Systems*, (K.J. Hammond (Ed.)), AAAI, pp. 341-346.

Zimmermann et al., 2002a — Zimmermann, J.U., Haasis, S., and Houten, F.J.A.M. van (2002). ULEO – Universal linking of engineering objects. In: *CIRP Annals 2002*, San Sebastian, Spain.

Zimmermann et al., 2002b — Zimmermann, J.U., Haasis, S., and Houten, F.J.A.M. van (2002). Applying universal linking of engineering objects in automotive industry – practical aspects, benefits and prototypes. *Proc. of DETC*, Montreal, Canada.

Appendices

Appendix A

Technological Product Specification

This chapter addresses technological product specification in order to provide the necessary background information for the technological DFs introduced. Perfect parts without any size or geometric deviation from nominal shape cannot be achieved. Variations of parts are caused by inaccuracies of manufacturing processes (Pivert and Rivière, 1999). Jorden (2001) describes the deviations of size, form, position, and surface using the example of the through hole. Figure A.1 depicts that the through hole may be too big or too small, curved, not straight or placed in the wrong position. Furthermore, the surface may be rough and scratched, which is usually referred to as deviation in the fine structure.

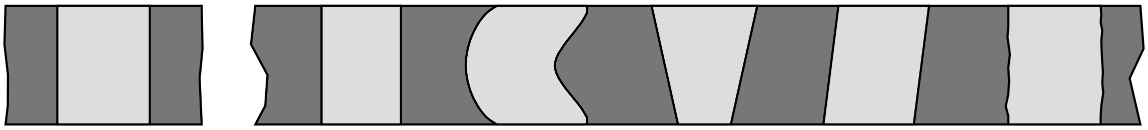

Figure A.1: Deviations from Nominal Shape (Jorden, 2001).

A set of requirements concerning shape, dimensions, surface characteristics and tolerances of a part or product is referred to as geometric product specifications (ISO/TR 14638, 1995). This thesis applies the term technological to refer to technological DFs as a subset of geometric product specification.

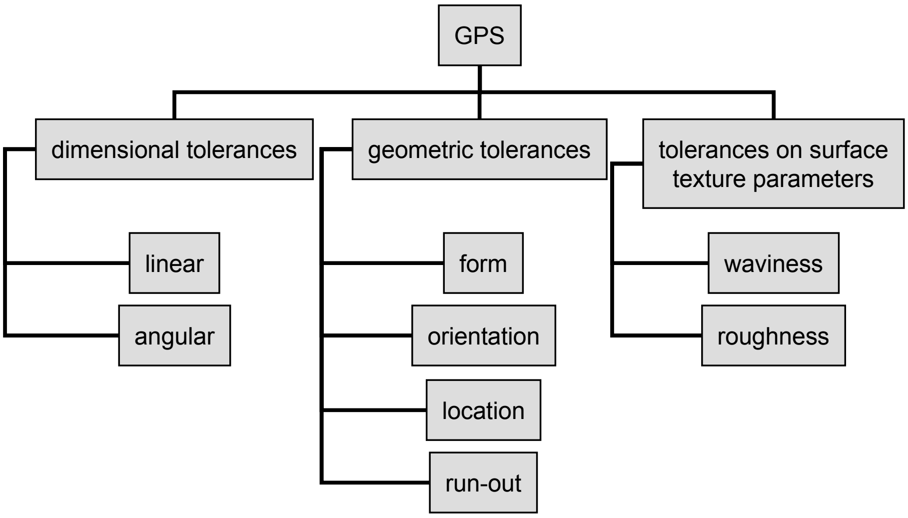

Figure A.2: Geometric Product Specification.

A.1 Tolerances

Since the functionality of a product depends on the actual size and actual geometric deviation of the parts to be assembled, the permissible part's deviation from its defined shape and dimensions has to be carefully and completely specified and controlled in order to guarantee the functionality of the product. Tolerances in general specify this permissible deviation. In tolerancing, the size, form, and position, i.e. dimensional and geometric tolerances are usually distinguished. Dimensional tolerances specify the allowance of size or distance variation (Tsai and Wang, 1999). The dimensional tolerances define the lower and the upper limits of a linear or angular dimension. In contrast, geometric tolerances limit the variation of shape or position of an element and are usually subdivided into tolerances of form, orientation and location, and run-out tolerances (ISO 1101, 1983; ASME Y14.5M, 1994). However, tolerances are crucial not only for the functionality but also for economy. In fact, they are one of the key cost drivers.

Tolerancing is one of the most important tasks in product development since tolerances directly impact two opposite objectives: quality and costs. Ideally tolerances are specified in such a way that optimal ratio between quality and costs is obtained. In tolerancing, especially in computer-aided tolerancing (CAT), the literature distinguishes primarily between tolerance representation, tolerance specification, tolerance synthesis, and tolerance analysis. Tolerance representation portrays how tolerances are represented inside a computer system. This refers to relating geometry and tolerances in the product model (Thome, 2001). It is therefore of great importance when tolerances are to be processed (Salomons et al., 1998).

Tolerance specification is the activity of defining tolerance types, their values, and, if necessary, datums (Salomons et al., 1998). It is preferably carried out in conformance with a tolerancing standard (Salomons, 1995). However, the standards do not prescribe how tolerances must be specified. Tolerance specification is still based on experience and trial-and-error methods (Tsai and Wang, 1999). This thesis applies the term tolerance modeling synonymously to tolerance specification.

The objective of tolerance synthesis is to optimize and complete the tolerance specification. In contrast, tolerance analysis aims at verifying the proper functioning of the assembly after the tolerances have been specified (Salomons, 1995). Generally two types of tolerance analysis are differentiated: worst-case analysis and statistical analysis (Mannewitz, 1994). In volume production, tolerancing is typically based on statistical techniques whereas, in small batch manufacturing, tolerancing is usually based on worst-case scenarios for assemblability (Houten and Kals, 1999).

There are a variety of national and international standards for the application and correct interpretation of tolerances. The key American standard related to dimensioning and geometric tolerancing is ASME Y14.5M–1994. This standard contains the pertinent information on geometric product specification. However, there is not just one comparable ISO standard but several standards to which the information is distributed. Dimensional tolerances are standardized in ISO 286–1988, geometric tolerances are specified in ISO 1101–1983, and general tolerances for mechanical parts manufacturing with metal-cutting methods are listed in ISO 2768-1–1989 and ISO 2768-2-1989. Equivalent standards also exist for other materials and manufacturing methods, e.g. for cast iron and polymer parts (Stark, 1994).The mutual interference and exclusion necessitates further standards:

- DIN 7167–1987: envelope requirement.
- ISO 8015–1985: principle of independency.
- ISO 2692–1988: maximum material principle.

With respect to the key characteristics ASME Y14.5M–1994 corresponds to ISO 1101–1983 (Stark, 1994). One of the few differences is that, in the ASME standard, the envelope requirement is generally valid (ASME Y14.5M, 1994; ISO 1101, 1983). In the remainder of this chapter the focus is set on the ISO standards.

The intention of the standards was that the interpretation of the specified tolerances be left to humans. Thus, the standards are driven by drawing, not preserving unambiguity in computer processing. For this reason, the standards mentioned above do not provide a tolerancing model which is consistent with 3D product models. Yet 3D CAD systems entail adequate tolerance modeling methodologies with the objective of tolerancing in 3D to provide a mathematically correct model for tolerancing. Houten and Kals (1999) state that a reliable methodology for tolerance specification is needed. Presently, most of the established CAD systems contain consistent methods for 3D tolerance specification.

Any tolerancing method that is in accordance with the international standards is more appropriate than one that is not. Geometric dimensioning and tolerancing (GD&T) represents such an appropriate method because it is consistent with ISO's and ASME's tolerancing standards. GD&T overcomes the ambiguity of purely dimensional tolerancing. In GD&T, the position of a geometric element, for example, is not toleranced using a dimensional tolerance, which would not prescribe the datum for inspection, but instead using a tolerance of position with a prescribed datum system.

For tolerancing applying ISO standards, there are two fundamental tolerancing principles to be considered (Jorden, 2001):

- Principle of independency (ISO 8015, 1985).
  Each specified dimensional or geometric requirement shall be met independently.
- Envelope requirement (DIN 7167, 1987).
  Unless otherwise specified, the envelope of perfect form at maximum material size of the geometric element shall not be violated.

The principle of independency and the envelope requirement are mutually exclusive. If necessary to ensure functionality, for the principle of independency the envelope condition has to be specified for each geometric element of size separately. The envelope condition is solely applied for cylindrical surfaces or sets of two opposed parallel surfaces and relates only to a single geometric element of size. Therefore, it does not comprise deviation of position. Both tolerancing principles require the envelope condition for the faces of fits. Although they have to be explicitly marked for the principle of independency, for the envelope requirement the envelope condition is valid for all simple geometric elements, i.e. cylinders and opposing parallel, plane areas.

According to Jorden (2001), the maximum material principle in ISO 2692–1988 is not a reasonable fundamental tolerancing principle since it can only be applied to discrete geometric elements, namely for the principle of independency and for the envelope requirement. Applying the maximum material principle, discrete tolerances of form can be exceeded as long as the sum of tolerances is maintained (actual size of a mating part does not reach the maximum material size) and functionality is guaranteed (ISO 2692, 1988).

As described, tolerances are divided into two main groups: dimensional tolerances and geometric tolerances. Dimensional tolerances are a specification of the allowable deviation of a dimensional value from its nominal value. They are either metrically toleranced and specified by an upper and a lower deviation referred to as plus-minus tolerance; or they are toleranced by applying tolerance classes (ISO 286-1, 1988). For the latter the nominal size, fundamental deviation, and size of the tolerance zone have to be specified.

In contrast , geometric tolerancing is based on toleranced elements, tolerance zones, and possibly datum elements or datum systems. Geometric tolerances are tolerances that are applied to control form, profile, orientation, location, and run-out (ASME Y14.5M–1994) and are either single or related. Examples for single tolerances are straightness, flatness, circularity, and cylindricity. The objective of geometric tolerances is to specify a boundary of spaces with regard to the datums specified or datum system in which the toleranced geometric element has to be located. The main benefit of geometric tolerances is that tolerances of position and form can largely be selected independently. In ISO 1101–1983 the geometric tolerances are defined as a zone in which the real element must be contained. The zone may have the form of a cylinder or a circle, space between two parallel planes or straight lines, space between two coaxial cylinders or circles, etc. (Humienny, 2001).

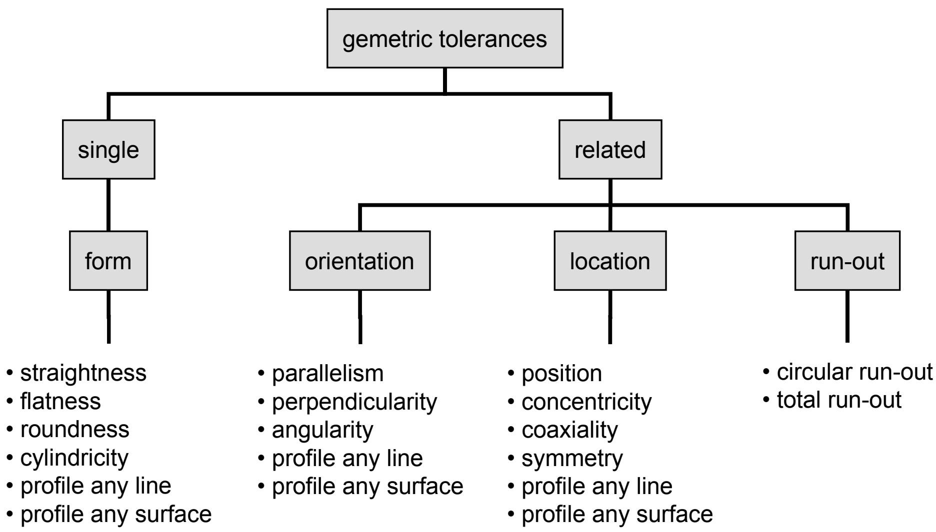

Figure A.3: Geometric Tolerances.

Tolerances of form, e.g. flatness, cylindricity, are individual tolerances and are not related to a datum. Tolerances which are related to a datum are tolerances of orientation (parallelism, perpendicularity, etc.), tolerances of location (position, concentricity, etc.), and run-out tolerances (circular and total run-out). Figure A.3 provides an overview.

A datum for related tolerances is a theoretically exact geometric reference, e.g. axis, plane, to which toleranced geometric elements are related (ISO 5459, 1981). Two or more separate datums used as a combined reference for a toleranced geometric element shape a datum system. A datum element is a real geometric element of a part, e.g. face, which is applied to generate the location of a datum. Tolerance zones and datums are geometrically perfect elements. Thus, tolerancing defines the association of elements of ideal form to real elements and describes the toleranced element (Mathieu et al., 1998).

Datums are established on one element (single datum) or on a set of elements. For a set of elements, the elements are either simultaneously (common datum) or sequentially (datum system) taken into account (Mathieu et al., 1998). In the case of a surface, the datum element may vary from its ideal form (ISO 5459, 1981). Datum targets are limited areas on the part applied to validate whether the datum element meets the requirements.

The elements of a part always have a size and a geometrical shape. Since the function of the part requires limitations to be set for the deviation of size and for the deviation of geometric characteristics such as form, orientation, and location, the tolerancing of the model should be complete. The application of general tolerances for size and geometry simplifies the task of ensuring that this requirement is met (ISO 2768-1, 1989). ISO 2768-1–1989 specifies the tolerances for linear and angular dimensions while ISO 2768-2–1989 specifies the geometric tolerances for elements, both without any individual tolerance indications. Both standards apply primarily to geometric elements which are manufactured using material removal manufacturing methods. There is another complementary standard: ISO 8062–1994 specifies the casting tolerances.

If tighter tolerances are required or larger tolerances do not endanger the function, such tolerances have to be explicitly defined in the model with local data given priority. The designation specifies the tolerance class (fine, medium, coarse, very coarse). These tolerance classes correspond to customary workshop accuracy. The permissible deviation is always symmetrical to the nominal measure, irrespective of whether it is a linear or angular dimension, but depends on the value of the nominal measure.

ISO 2768-2–1989 specifies general geometric tolerances in three tolerance classes. If smaller geometric tolerances are necessary, these tolerances have to be indicated directly in accordance with ISO 1101–1983. General geometric tolerances apply to the following geometric tolerance characteristics: straightness, flatness, circularity, parallelism, perpendicularity, symmetry, coaxiality, circular run-out. In each case there are three tolerance classes.

A.2 Surface Texture

In the previous section the focus was placed on deviations in the coarse structure. In this chapter the emphasis lies on deviations in the fine structure as it is impossible to manufacture parts without any deviation not only from the nominal shape but also from the nominal surface, i.e. the surfaces of parts are not perfectly smooth and thus have to be quality controlled.

The manufacturing methods applied will cause marks on the surface – referred to as surface texture – which consists of a series of peaks and valleys (Humienny, 2001). Conventionally the texture is made up of characteristics defined as waviness and roughness. These undulations, grooves, and score marks together with the deviation of shape usually superimpose on the actual surface (DIN 4760, 1982). DIN 4760–1982 breaks the surface down into six categories and form, waviness, and roughness (this term is applied for the third and the fourth class) are designated as the first, second, third, and fourth classes of profile deviation.

Waviness may be caused by eccentric clamping, vibrations of the machine tool, tool chatter, etc. Roughness is an irregularity inherent in the production process left by the machining agent such as cutting, tearing, and surface fatigue (Humienny, 2001). Surface imperfections – pores, scratches, etc. – as defined in ISO 8785–1998 are out of the scope of this thesis and therefore not considered any further.

As for tolerancing, there are also several national and international standards for the correct application and interpretation of surface textures. The main US standard is ASME B46.1-2002. The focus in the remainder of this chapter is chiefly on ISO standards.

The international standard ISO 1302–2002 specifies the rules for the indication of surface texture in product models. It distinguishes between these basic surface requirements:

- Any manufacturing process allowed (APA).
- Material removal required (MRM).
- No material removed (NMR).

In addition to this indication, in order to ensure unambiguity, complementary requirements can be specified. These are four in all:

- One or more surface texture requirements.
- Manufacturing method.
- Surface lay and orientation.
- Machining allowance.

The indication of the surface texture requirement includes the type of surface profile (primary profile, waviness profile, and roughness profile), the characteristic of the profile, the number of sampling lengths making up the evaluation length, and the way the specification limit has to be interpreted (ISO 1302, 2002). Figure A.4 depicts the structure of the surface texture indication.

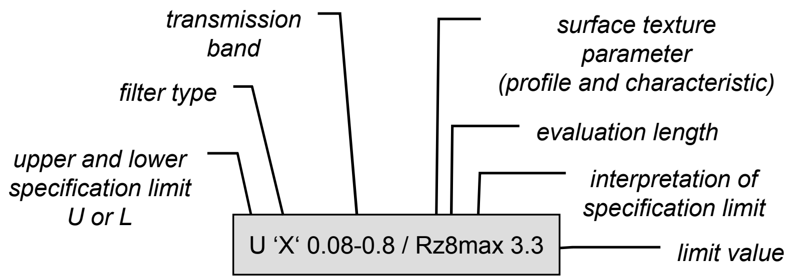

Figure A.4: Indication of Surface Texture.

The last two chapters have described tolerances and surface texture. Both significantly influence manufacturing and resource consumption as will also be shown in more detail in appendix B. Therefore, these requirements have all to be taken into account when interpreting the CAD product model for design-concurrent cost estimation.

Appendix B

Tolerances and Their Impact on Costs

The activity of specifying tolerances can have a large impact on quality and costs. On the one hand, it affects the function of the final product, on the other hand, tolerance requirements determine the selection of manufacturing and inspection methods, machines, tooling, fixtures, scrap, rework, etc. The tighter the tolerance of a geometric element is, the higher its manufacturing costs are because feeds may be reduced, the number of passes increased, or an intrinsically more accurate process chosen, all requiring more time. In short, tolerances decisively affect the production costs.

Jorden (2001) terms the relationship between manufacturing costs and the tightness of the tolerance of a geometric element the cost hyperbole and describes it using

$$K \sim \frac{1}{T^{0.8..1}} \tag{19}$$

with *K* corresponding to the costs and *T* to the tolerance. Jorden (2001) restricts the validity of the equation to the application of the same manufacturing method within its economic section and the dimensional tolerances. The real course of the cost-tolerance curve has steps. Houten and Kals (1999) state that relations between the tightness of tolerances and the manufacturing cost are usually non-linear and show discontinuities when the limits of process capabilities are exceeded. Figure B.1 depicts an example.

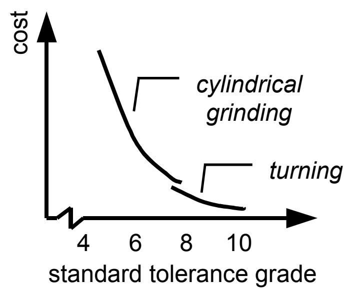

Figure B.1: Discontinuities of Cost-Tolerance Curves.

Chase (1999) sketches a literature review on cost-tolerance models as depicted in the table below. The constant coefficient ***A*** represents fixed costs and may include setup cost, tooling, material, prior operations, etc. The term ***B*** denotes the cost of manufacturing a single dimension to a specified tolerance. The exponent ***k*** describes how sensitive the costs are to changes in tolerance specification. The reciprocal cost model of Chase and Greenwood is similar to the relationship described by Jorden (2001).

cost model	function	author
reciprocal squared	$A+\frac{B}{tol^2}$	Spotts
reciprocal	$A+\frac{B}{tol}$	Chase and Greenwood
reciprocal power	$A+\frac{B}{tol^k}$	Chase et al.
exponential	$A\cdot e^{-B\cdot tol}$	Speckhart

Table B.2: Cost-Tolerance Models in Literature (Chase, 1999).

The multitude of cost-tolerance models in literature makes clear that depending on specific circumstances, e.g. materials, resources, the course of the curve differs strongly (Nusswald, 1998).

Applying the example of drilling as portrayed in Jorden (2001), drilling applying a twist drill achieves a standard tolerance grade of IT 12. Thus, the hole does not become cheaper if H14 is specified. However, if H10 is required, instead of drilling, drilling in combination with reaming is necessitated. The costs of the operation rocket, tripling, and the quality of the hole becomes H7.

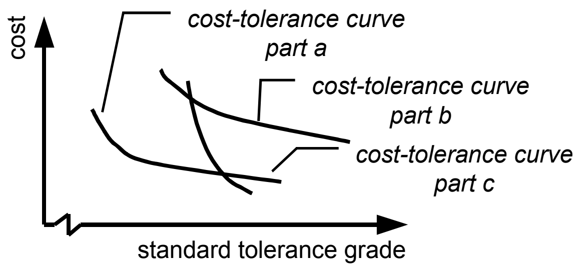

Figure B.3: Cost-Tolerance Curves (Chase, 1999).

Chase (1999) presents cost-tolerance curves for parameters of parts of an assembly as depicted in Figure B.3 Since they differ, the curves are applied to reduce the costs of the assembly by increasing the tolerances of the parameters which significantly influence the costs and decrease the tolerances which only slightly impact the costs in order not to violate the assembly tolerance. This of course necessitates the presence of the cost-tolerance function. As manufacturing cost data are usually not published since they represent trade secrets, little verified information is available concerning cost-tolerance relationships. Until a mathematical model is available, optimization of tolerances with regard to costs is not viable. Empirical functions that describe the relationship between tolerance and cost are needed. But cost-tolerance data is very scarce. Companies consider it proprietary so it is mainly not published.

Index

Abbreviations

AI .... artificial intelligence
ANN .... artificial neural network
CABACO .... case-based cost estimation
CAD .... computer-aided design
CAM .... computer-aided manufacturing
CAPP .... computer-aided process planning
CAT .... computer-aided tolerancing
CAx .... comprehensive term for all computer-aided technologies
CBR .... case-based reasoning
CCA .... cost center accounting
COA .... cost object accounting
CTA .... cost type accounting
DF .... design feature
EO .... engineering object
EOR .... engineering object relation
ERP .... enterprise resource planning
FLC .... front load costing
GD&T .... geometric dimensioning and tolerancing
GE .... geometric element
GEOR .... generative engineering object relation
GT .... group technology
IEOR .... informational engineering object relation
IF .... inspection feature
IR .... integration relation
KADS .... knowledge analysis and design support
KBS .... knowledge-based system
KE .... knowledge engineering
MF .... manufacturing feature
MTRT .... meta taxonomy of relation types
NC .... numerical control
OEM .... original equipment manufacturer
PART .... planning of activities, resources, and technology
PART-S .... planning of activities, resources, and technology – sheet metal
PCP .... product creation process
PDM .... product data management
PPR .... product, process, and resource
ProSAp .... process step application
SE .... structuring element
ULEO .... universal linking of engineering objects
UMEO .... unified model of engineering objects

Research of the OPM Laboratory

Research carried out within the Laboratory of Design, Production, and Management embraces the manufacturing of industrial products and is focused on developments in computer-aided manufacturing, covering the overall range from design to the integral control of the activities on the shop floor. Over the last decade, the integrated approach towards the product realization process has become a necessity because – for various reasons – industry has increasingly been experiencing problems with the implementation of part-solutions. Reports of the research projects are distributed in a limited edition by the Laboratory of Design, Production and Management. The series is published with the ISSN number 1386-5307. Dissertations that have been released previously are listed below:

A.H. van ‘t Erve	Generative computer aided process planning for part manufacturing, an integrated approach (1988).
J.R. Boerma	The design of fixtures for prismatic parts (1990).
F.J.A.M. van Houten	PART: a computer aided process planning system (1991).
J.J. Tiemersma	Shop floor control in small batch part manufacturing (1991).
F.J.C.M. Jonkers	A software architecture for CAPP systems (1992).
H.J.W. Vliegen	Classification systems manufacturing; managerial control of process knowledge (1993).
A. Lenderik	The integration of process and production planning in small batch part manufacturing (1994).
L.J. de Vin	Computer aided process planning for the bending of sheet metal components (1994).
R.M. Boogert	Tool management in computer aided process planning (1994).
O.W. Salomons	Computer support in the design of mechanical products (1995).
A.L. Arentsen	A generic architecture for factory activity control (1995).
R. Geelink	Flexible definition of form features (1996).
J. de Vries	Integrated process planning for small batch manufacturing of sheet metal components (1996).
J.H. Kappert	Integration of component design, process planning and die design in rubber pad forming (1997).
A. Liebers	An architecture for cost control in manufacturing (1998).
R.E. Begelinger	Computer support in the design of product families (1998).
M.M.T. Giebels	EtoPlan; A concept for concurrent manufacturing planning and control (2000).
D. Lutters	Manufacturing integration based on information management (2001).
T.H.J. Vaneker	Development of an integrated design tool for aluminum extrusion dies (2001).
S. Finke	Solid freeform fabrication of metal components by extrusion and deposition of semi-solid metals (2002).
E. Ten Brinke	Costing support and cost control in manufacturing (2002).
D. Wijnker	Integration of information in manufacturing systems (2003).

"I've seen the future and it's much like the present, only longer." Dan Quisenberry, pitcher