Designing and manufacturing assemblies

Citation for published version (APA):
Net, van der, A. J. (1998). *Designing and manufacturing assemblies*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR513090

DOI:
10.6100/IR513090

Document status and date:
Published: 01/01/1998

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 04. Oct. 2023

Designing and Manufacturing Assemblies

A.J. van der Net

Designing and Manufacturing Assemblies

Cover: Design by B. Mobach and A.J. van der Net.
The cover shows an illustration from *De Re Metallica* by Georgius Agricola (1494-1555). This work, of which the title literally translates as *On the Nature of Metals*, reviews everything about mining and is regarded as the first complete dissertation in the field of mining and metallurgy.
The illustration is taken from *Book VI, Miners' tools and machines*, describing the machinery used in mining. This particular engraving, showing fans used for blowing air into mining shafts, is one of the first known assembly drawings.

Print: Universiteitsdrukkerij Technische Universiteit Eindhoven

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Net, Antonius J. van der

Designing and manufacturing assemblies / by Antonius J. van der Net. -
Eindhoven : Technische Universiteit Eindhoven, 1998.
Proefschrift. - ISBN 90-386-0780-6
NUGI 841

Trefwoorden: CAD/CAM; assemblage / werkvoorbereiding / fabricagegericht ontwerpen / toleranties
Subject headings: CAD/CAM; assembly / process planning / design for manufacturing / tolerances

Designing and Manufacturing Assemblies

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de Rector Magnificus, prof.dr. M. Rem, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen op
donderdag 3 september 1998 om 16.00 uur

door

Antonius Johannus van der Net

geboren te Tilburg

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J.E. Rooda
en
prof.ir. D.C. Boshuisen

Preface

This thesis has resulted from a research project at the Faculty of Mechanical Engineering at the Eindhoven University of Technology. The project concerned the integration of design and manufacturing. I had a great deal of pleasure working on it. I hope the results contribute to a better understanding of these complex processes.

The work described in a thesis is always the result of a collaborative effort. A lot of people spent time and effort or contributed otherwise to the realisation of this thesis.

First of all, I would like to express my gratitude to professor J.E. Rooda for giving me an opportunity to complete my work in the Systems Engineering group, for his support, and for his comments on my thesis. I also would like to thank professors D.C. Boshuisen, P.H.J. Schellekens, and P. de Ruwe for their comments.

I thank all my former and present colleagues, for contributing to this work as well as for providing an enjoyable working environment. I am not going to present a long list of names, but I sincerely give thanks to all roommates, staff members, students, web team enthusiasts, coffee corner companions, representatives of the Stan Ackermans Institute, and all others I have worked with in the past years.

Last but not least I want to mention the 'home front', my family and friends, who sometimes had a hard time explaining why I was still 'in school', but never stopped believing in a successful outcome. They expressed their confidence by quoting the South African phrase 'Moenie worry nie, alles sal reg kom'. Obviously, they were right.

Summary

Design is one of the most important activities in creating a product. It needs to be supported as designers face short design cycles and an increasing amount of information. Current product modelling tools and design methods only partly provide support. Tolerances, which are of importance for functionality and largely determine manufacturing costs, receive little attention.

Systems theory and life-cycle concepts put the design and manufacturing activities into perspective and point out the relations between a product and its production system. An important issue is the influence of the design process. By preventing errors to appear in designs, lead times are reduced, added value is increased, and process planning can be automated.

A model of the product creation process, which represents the activities that transform requirements into products, is used to explain an approach that enables manufacturability evaluation during design. Every data element or decision added by a designer is evaluated for manufacturability. A product model results that can be manufactured right first time. This evaluation can be automated, but is limited to geometrical design and a certain type of products. It also results in a less strict separation between design and process planning activities. Product modelling is based upon the states and state transitions approach, while evaluation is based upon comparing factory capability with product requirements.

When modelling an actual product, geometrical relations, assembly relations, and tolerance relations are of importance. The design primitives used in the relations are cylindrical faces, planar faces, spheres, cones, and lines. Because the relations are not really convenient to work with, complex primitives are introduced. To prove that the concepts are consistent and unambiguous, they were described as elements of a formal specification language, which is also used for implementation purposes.

The design operations are mapped upon manufacturing operations, with a focus on assembly operations. Assembly process planning deals with the generation of manufacturing information needed to transform a group of loose parts into an assembled product. A part can be assembled using four operations: feed, grasp, move, and mount. Each of these operations is separately checked for validity to make sure an assembly is manufacturable. Besides that, a suitable assembly sequence is found. Inspection planning is also an integral part of manufacturing planning, so checks are incorporated for guaranteeing that a part can be measured properly as well. As in design, the concepts used in manufacturing planning can be expressed in a grammar.

A case study is used to validate the concepts presented. The design method using relations and primitives is well suited for designing an actual product, but strongly depends on well-defined primitives and feedback to the designer. The tool forces the designer to find out what dimensions are essential to functionality, but the absence of quantitative support in tolerancing devaluates the merit of the current tool. A lot of effort is needed to fully implement the tool and the information databases needed.

It is recommended to create a fully functional implementation. The design language presented is a suitable specification and should be developed further. Besides that, a lot of work is to be done in finding relations between functionality and tolerances and in establishing an interface between the design support tool and other tools.

Samenvatting

Het ontwerpen, een van de belangrijkste activiteiten in het voortbrengen van producten, dient te worden ondersteund, aangezien ontwerpers te maken hebben met zeer korte ontwerptijden en een toename van informatie. De op het moment beschikbare gereedschappen en methoden kunnen slechts ten dele aan deze vraag om ondersteuning voldoen. Aan toleranties, van belang voor het functioneren van producten en voor een groot deel bepalend voor de fabricagekosten, wordt te weinig aandacht besteed.

De systeemtheorie en de levenscyclus worden gebruikt om de activiteiten rondom ontwerpen en fabriceren in een groter verband te plaatsen en om de onderlinge relaties van een product en een productiesysteem aan te geven. De invloed van het ontwerpproces is van groot belang. Door het voorkomen van fouten in een ontwerp kan de doorlooptijd worden verkort, de toegevoegde waarde worden vergroot en de werkvoorbereiding automatisch worden uitgevoerd.

Een model van het productcreatieproces, wat de activiteiten weergeeft die nodig zijn voor het creëren van producten, wordt gebruikt om de aanpak die het evalueren van fabriceerbaarheid mogelijk maakt te verduidelijken. Elk data-element of elke beslissing die de ontwerper toevoegt wordt gecontroleerd op fabriceerbaarheid. Het resultaat is een productmodel wat meteen zonder problemen gefabriceerd kan worden. De evaluatie kan worden geautomatiseerd, maar dit beperkt zich tot het geometrisch ontwerpen van een bepaald type producten. De scheiding tussen activiteiten op het gebied van ontwerpen en werkvoorbereiding zal minder duidelijk worden. Het modelleren van producten is gebaseerd op het beschrijven van toestanden en toestandsovergangen. De evaluatie is gebaseerd op het vergelijken van de fabrieksprestatie met de eisen beschreven in het productmodel.

Wanneer een product wordt gemodelleerd, zijn geometrische relaties, assemblagerelaties en tolerantierelaties van belang. De primitieven die in deze relaties worden gebruikt zijn cilindrische vlakken, platte vlakken, bollen, kegels en lijnen. Omdat met relaties niet echt makkelijk te ontwerpen is, worden hierbij meer complexe primitieven gebruikt. Om aan te tonen dat het geheel consistent en eenduidig is, zijn

alle elementen beschreven als deel van een formele ontwerptaal. Deze taal is ook geschikt als specificatie voor een implementatie.

De ontwerpoperaties worden afgebeeld op fabricage-operaties, waarbij de nadruk ligt op assemblage. De werkvoorbereiding voor assemblageprocessen genereert de informatie die nodig is om een verzameling onderdelen om te zetten in een samengesteld product. Het assembleren van een onderdeel gebeurt door de operaties toevoeren, grijpen, verplaatsen en invoegen. Elk van deze operaties kan los van de andere worden gecontroleerd op realiseerbaarheid om zeker te kunnen zijn van fabriceerbaarheid van de samenstelling. Daarnaast wordt ook een geschikte assemblagevolgorde bepaald. Omdat het plannen van het meten van producten onderdeel is van de fabricageplanning, wordt ook dit geëvalueerd. Analoog aan ontwerpoperaties kunnen fabricageoperaties worden beschreven in een grammatica.

De gepresenteerde concepten zijn gevalideerd door middel van een case study. De ontwerpmethode gebaseerd op relaties en daaruit samengestelde primitieven blijkt te voldoen bij het ontwerpen van een product, maar is sterk afhankelijk van goede primitieven en terugkoppeling naar de gebruiker. Het gebouwde gereedschap dwingt de ontwerper aandacht te besteden aan de dimensies die van belang zijn voor de functie, maar het gebrek aan kwantitatieve ondersteuning doet afbreuk aan de verdienste van het huidige gereedschap.

Het is aan te bevelen om allereerst een volledig functionerende implementatie te maken. De formele ontwerptaal kan daarbij als specificatie dienen en dient verder te worden ontwikkeld. Daarnaast dient aandacht te worden besteed aan het vinden van relaties tussen functionaliteit en toleranties en het koppelen van het gepresenteerde ontwerpgereedschap aan andere gereedschappen.

Contents

Chapter 1

Introduction

Man has always designed things to make life comfortable. Design involves organising creative and analytical processes to satisfy needs or to solve problems. Design problems have a goal, constraints within which the goal must be achieved, and criteria by which a successful solution is recognised. It covers the refinement of existing artifacts as well as conceiving new artifacts. Designers use creativity and knowledge about physical effects and working principles to find concepts that may offer a solution to a problem. Analysis is needed to find out more about the characteristics of a concept, that is, to prove it is suitable. A designer also needs to posses knowledge about manufacturing technologies. The complexity of the design process gives rise to a need for concepts and tools that aid a designer in creating good designs. This concerns design methodologies and tools for analysis, optimisation, and manufacturing preparation.

An essential design activity is the generation of a description of the artifact, as the solutions generated should be communicated to others in an understandable form. Creating a model of the artifact is a means to gather and structure knowledge. A model is used to analyse a problem, for communication, and to perform experiments or simulations. The design process evolves around creating a suitable representation of the artifact being designed: a product model. Traditionally, product models are based on product geometry (the engineering drawing or its electronic equivalent).

1.1 A history

Artisans used to design as well as manufacture a product; product models existed in their mind. To increase performance, design and manufacturing activities were separated, as proposed by Adam Smith in the 18th century. Joshua Wedgwood was one of the first to use this separation in large scale design and manufacturing of products. The result was a rising need for communication, which in design is mainly graphically based [Bert95]. Since the first known technical drawings (2130 BC), the models evolved through concepts like co-ordinate geometry (Descartes), descriptive geometry (Monge), and isometric drawing (Farish) [Book63]. The computer has had a major impact on the creation of models. The Sketchpad system (Sutherland, 1962) is considered the beginning of Computer Aided Design (CAD). Since then, CAD is closely linked to the capabilities of computer hardware. Commercial applications emerged as the cost of hardware decreased while performance increased. Standards such as IGES and STEP (ISO 10303) enable the exchange of models.

Developments in product modelling relate to markets, competitors, and technologies and should increase the performance of the design process. The changing environment has led to a change in performance criteria. Markets have changed from bidders' markets to buyers' markets, companies operate in a global economy, and the pace of technological developments has increased [Bosh92, Feld96]. The evolution of performance criteria also had an impact on the design process. Efficiency, expressed in the price of products, was traditionally the most important criterion. During the 1960s, quality became important to the customer as well. In the 1970s, customers also require a product that is up-to-date. More individualised consumers demand ever changing series of products in the 1980's and 90's. The coloured bars in Figure 1.1 depict the rise of these criteria [Bolw91]. A fifth market demand currently emerging is the demand for products that are less a burden to the environment [Brez96].

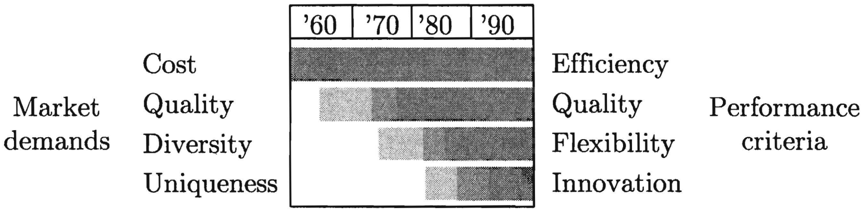


Figure 1.1: Evolution of market demands

Manufacturers compete to be the first to market customised products having sufficient quality and a competing price. This requires a great product range, short product runs, reduced lead time, rapid turnover, and a small stock of partly finished and finished products. Designers need to provide product designs in short design cycles and face a considerable increase of information. A significant part of the competitive advantage is obtained by minimising the lead time of an order. The lead time is the time needed to create a product, including design. This calls for short design cycles, rapid process planning, and fast manufacturing and shipping. The objective is not to finish the manufacturing operations as quickly as possible, but to minimise the time needed for designing and manufacturing a product. An extension of tasks in design and process planning is therefore required.

1.2 State of the art

This paragraph discusses the state of the art in design support; methodologies, tools, and representations. Computer systems are dominantly present in today's design process (Computer Aided Design, Computer Aided Manufacturing, Computer Aided Engineering). Some of the techniques used in these systems are assessed.

Geometric modelling

Geometric modelling creates a valid computer representation of an object's form to communicate, analyse, and visualise a design process [Fing89b, Shah95]. Geometric modellers have existed for 25 years and have undergone vast changes. Earlier modellers were based on Wire Frame Representation, in which curves represent the edges of a physical object. This evolved to solid models, that are able to unambiguously model a volume. The most important solid modelling representations are Boundary Representation [Brai74] and Constructive Solid Geometry [Voel77], depicted in Figure 1.2. Boundary representation describes an object in terms of its topological boundary (faces, directed edges, vertices). Constructive solid geometry represents it as a series of Boolean operations on simpler solids (cubes, spheres, cylinders). Both representations allow manipulation of volumes and Boolean operators.

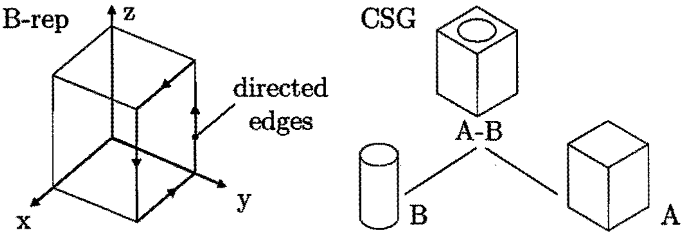


Figure 1.2: Solid modelling representations

Over the years, designers have been provided with increasing capabilities to create the desired geometry. Operations such as blending an edge, extruding a face, intersecting two solids, and creating doubly curved surfaces are available. The output of the modelling process however, is typically the final geometry only. Often this final geometry is used to create engineering drawings. Dimensions and tolerances have to be added to the drawings manually. The drawings are used for further processing, for instance manufacturing or communication with clients or suppliers.

A mismatch in abstraction level limits the use of geometry when modelling a mechanical design [Shah88, Arba89, Shah95]. The engineering meaning is expressed using entities such as lines, vertices and faces. Important information such as product structure, tolerances, and material properties is often missing or stored as an attribute only. This causes problems when the model has to be mapped onto physical reality. Besides that, geometric modelling renders insufficient information to complete other engineering activities, such as manufacturing preparation or analysis. The use of advanced geometric modelling tools has mainly caused an alienation from the manufacturing phase, which results in higher production cost and increased lead times. Sometimes the resulting geometry can only be created using rapid prototyping techniques, such as stereo-lithography, selective laser sintering, or laminated object manufacturing. Appendix A elaborates on these techniques.

Geometric modelling makes it easier to create and modify representations of a design. However, the mismatch in abstraction level and problems in manufacturing have instigated a search for entities that represent the functionality behind geometry. These entities should have an affinity with engineering activities in design, manufacturing, and analysis. Preferably, the entities are a natural form of communication [Nnaj90]. One of the results of this search is found in the concept of feature based modelling.

Feature based modelling

In mechanical design, features were proposed to serve as a means to raise the level of abstraction. A feature is an abstract concept, further defined in the context of a specific activity [Arba89, Bron93, Shah95]. Because of the multitude of existing activities, different interpretations of the feature concept have emerged, which have resulted into applications used for:

- modelling product geometry and assemblies
- generating manufacturing information
- analysis and optimisation
- representation of tolerances

Some definitions of features relate to their abstract nature, as demonstrated by the definitions of Brown and DeFazio respectively [Brow92, DeFa93].

> *A feature is any geometric or functional element or property of an object useful in understanding its function, behaviour or performance.*
>
> *A feature is any geometric or non-geometric attribute of a discrete part whose presence or dimensions are relevant to the product's or part's function, manufacture, engineering analysis, use, etc., or whose availability as a primitive or operation facilitates the design process.*

Most feature based applications use less abstract feature definitions: form features or manufacturing features. Form features are groups of geometric entities that define attributes of a parts' nominal size and shape [Shah88]. They are groups of geometric entities that form a recognisable shape. If decomposed, they reduce to meaningless geometric entities such as lines, points, and surfaces. Form features may or may not by themselves have a functional purpose. If the engineering meaning is not complex, it is represented by a single form feature, such as a hole. To capture more complex functionality, a composition of multiple form features is used, like the air cylinder mounting feature depicted in Figure 1.3 [ElMa93].

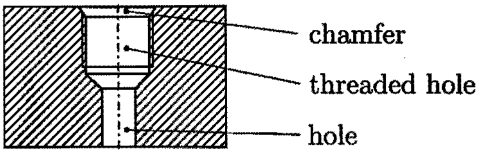


Figure 1.3: A composition of form features

Groups of geometric entities can also correspond to machining operations, so-called manufacturing features. Manufacturing features were proposed as a solution to the problem of automating process planning. They have their roots in the physics of manufacturing processes. Manufacturing features are often specified for material removal processes, such as milling, drilling, and turning. Figure 1.4 depicts some milling features [Krik92]. Examples of turning features are found in [Brow92].

Features require knowledge of the context or application domain in which their geometry has a meaning. Relevant features and relationships between those features are derived from the context or application domain. To create product models based on a specific context, features can be used as building blocks. Design by features is an attempt to design in terms of the interpretation, where features are primitives of the

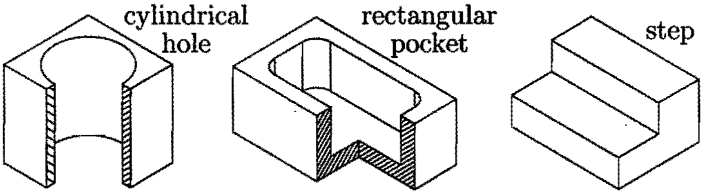


Figure 1.4: Manufacturing features

design process. The focus of these feature based applications is on parts rather than assemblies. Because of its context-sensitivity, design by features restricts the freedom of modelling. In many cases this is what is intended, as the limitations imposed by the modelling tools result in a certain type of product model. This is demonstrated by the number of applications using manufacturing features. Modelling with manufacturing features opens up the opportunity to incorporate manufacturing restrictions in design and to automatically generate manufacturing information [Delb89, Krik92]. This approach has proven to reduce lead time and costs. Most applications of design by manufacturing features are found in the field of material removal techniques.

Creating a product requires evaluation of the model in multiple domains. It is very difficult to provide product models with sufficient information concerning all domains. Although the definition of a feature suggests it contains any kind of relevant data, in current practice it will not carry all information needed. This would result in an amount of data that becomes unmanageable for the designer. If analysis of the product model is needed in a domain different from the one in which the product model was created, (re)interpretation is required. For interpretation of a product model, two options are available: feature mapping and feature recognition [Shah95].

In feature mapping, features used to create a product model are transformed into those of a required domain. Figure 1.5 shows an example. A protrusion feature (design domain) is mapped upon two milling features (manufacturing domain) [Vrie96]. Problems may occur in feature mapping as features from different domains are of different types, with different application rules and interactions. Sometimes the interactions determine the new features. In the example, the dimensions of the stock material determine the material to be removed and thus the parameters of the manufacturing features. Besides context sensitive, feature mapping is usually ambiguous.

Feature recognition extracts features from the geometry of a completed design. It assumes the information required is available in the geometric model, but not in a suitable format [Erve88]. Feature recognition needs the definitions of the char-

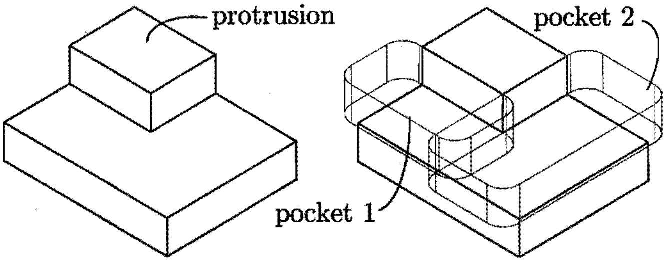


Figure 1.5: Feature mapping

acteristic shapes and parameters. An algorithm tries to find groups of geometrical entities that meet these definitions and extracts corresponding parameter values. Feature recognition is mainly used to extract manufacturing features for automated process planning, like in the PART system for material removal [Erve88, Hout91] or the DIAC project [Mart91] for assembly relations. A fundamental objection to feature recognition is that it uses the description of the final state of a design only, so information from the design process is lost. It is directed towards extracting an interpretation from a design that may have been explicitly intended by the designer [Brow95]. Besides that, it may not be possible to find matching features for all geometric entities, as the description was created in a different domain. It is difficult to use feature recognition for product model analysis during design, as the entire geometry needs to be analysed after each change.

Functional modelling

Geometric modelling techniques typically represent the nominal shape of a product, while feature based modelling techniques create a model based on a specific context. Both techniques have difficulties capturing the designers' view on the function of the part; the design intent. This complicates the re-use of designs and may lead to non-optimal performance of the product [Mänt91, Huus95, Shah95].

Functionality is composed of numerous separate sub-functions, each associated with their own abstraction of the final artifact and part of the geometry [Hua92]. The relationship between functionality and geometry is complex. Designers specify the functions of subsystems and how they interact, but still tend to use geometric abstractions during design. The major barrier in functional modelling is the lack of computational characterisation of the design intent in terms of mechanical functions, constraints, optimisation criteria, and other properties [Liba88, Arba89, Hash93]. Some systems exist to aid a designer in modelling functional structures. However, these systems usually have problems in connecting this functional structure to a geometrical model [Kutt93, Requ96]. Applications that incorporate some functionality are parametric design systems and knowledge based engineering systems.

In parametric design, the structure or attributes of the artifact are known at the start of the design process. Values are assigned to attributes that are called parametric design variables, which are numerical values as well as type or class designations. If a criterion function can be found, optimisation models can be used [Fing89a]. Currently, parametric design systems mostly use only parametric geometry. Computer Aided Design systems such as Unigraphics, Pro/Engineer, Solid Works, and Solid Edge enable a designer to specify parametric geometry and relations, which enables fast redesign through propagation of changes.

Closer to the original definition of parametric design are knowledge based engineering systems, like ICAD and Design++, in which models are built of products or processes by writing rules that describe the engineering process. When product requirements change or new versions are desired, the model is evaluated and a new design is created automatically [Kess89]. Knowledge based engineering tools capture and maintain product and design knowledge [Huus95]. As models contain algorithms describing specific products or processes rather than general knowledge, knowledge based design systems are better referred to as algorithmic design systems. Applications of algorithmic design lie in the field of engineering products with lots of interdependencies, many or complex iterations, or involving multiple engineering disciplines. Examples of such products are the turbine blades of a jet engine [Kess89] and a power plant [Huus95]. The limited use of algorithmic design systems is due to the unconventional way of working and the initial costs [Groo93b]. Creating the complex set of rules describing products and design knowledge requires a lot of time and effort. Usually, the knowledge is not readily available and difficult to formalise as it is based on experience and relies on the memory of designers.

Assembly

Assembly processes are very important, as they take up a significant part of the lead time and manufacturing costs. The focus of assembly research is mainly on assembly sequence, reduction of lead time, and development of flexible assembly equipment (including control systems). Other fields of attention are sensors, grippers, fixtures, transport, and compliance. The representation of assembly information in product models and manufacturing operations for assembly receive less attention.

The assembly operations needed are defined by part geometry, tolerances and relations between parts. The knowledge of assembly processes, especially their relation with product properties, is rather poor. Techniques like Design For Assembly [Boot82] are based on empirical data and aim at redesign of mass products. Product family design, used to improve the performance of automated assembly systems, is not based upon fundamental knowledge of assembly processes.

Design support for assembly recently starts to receive attention. Feedback to a designer on assembly operations however, lags far behind the support provided for the design of parts. Manufacturing features for assembly for example, are seldom found, probably because of a lack of knowledge about the relation between geometry and assembly processes. Besides that, the availability of well documented general purpose equipment for assembly (with standard processes) is limited.

Design models

Besides models of the artifact being designed (product models), models of the design process exist: design models. Design models provide a framework for the activities that occur in designing products. They are usually represented in flow diagrams, depicting the stages in design, the sequence of these stages, and feedback loops [Cros89, Ullm89, Fing89a, Suh90, Bles94, Rooz95]. Two types of models exist:

- Descriptive models describe the sequences of activities that occur in designing.
- Prescriptive models prescribe better or more appropriate patterns of activities.

An example of a descriptive model is the conceptual design model by French, depicted in Figure 1.6. It considers solutions on a conceptual level. The concepts are analysed and a solution is chosen, without trying to find all possible solutions for all sub-functions. Prescriptive models encourage a more systematic, algorithmic type of design. They can be considered design methodologies, as they emphasise the analytical work. Their purpose is to completely understand the problem and to prevent that any important element is overlooked. Examples are the models of Archer, VDI 2221, and Pahl and Beitz. The latter is also depicted in Figure 1.6.

Design models consider design as a sequence of activities that goes from a high abstraction level to details: top-down design. Although feedback from downstream activities is allowed, a designer starts with the main function of the product. This function is subdivided into sub-functions, until the smallest sub-functions are materialised individually. In bottom-up design, the designer perceives the entire product functionality. From the functions the product has to perform the designer generates some idea of the subsets of parts performing one or more of the desired functions. Detailed modelling of the product starts at the lowest conceptual level.

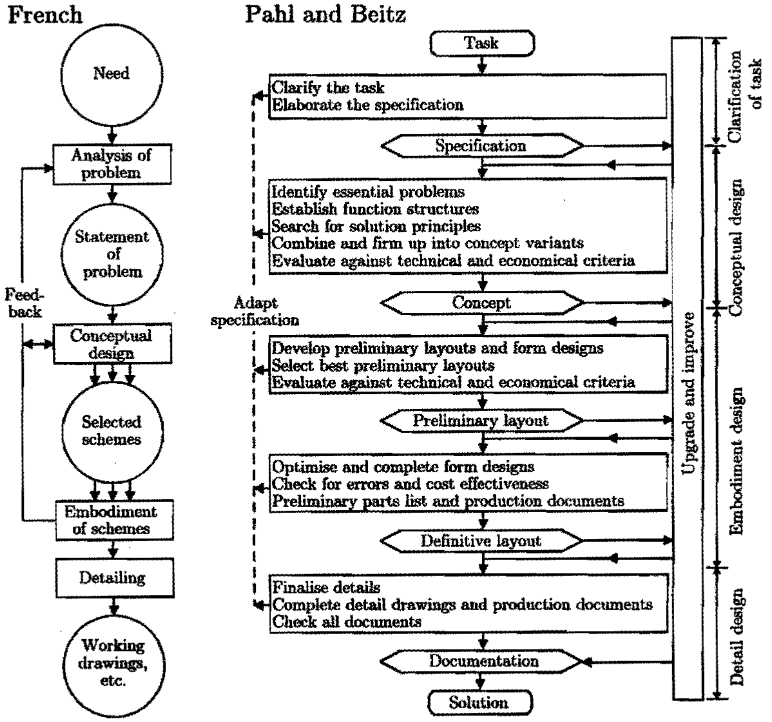


Figure 1.6: Models of the design process

A top-down approach is closest to the actual design process. However, top down design strategies encounter some problems, because a unique coupling from function to physical implementation is seldom found. Materialising separate functions may also cause sub-optimisation. Besides that, downstream knowledge is used to create good designs, such as available materials and limitations of manufacturing processes. On the other hand, bottom-up design needs some idea about the functionality of the parts, so it is not really bottom-up. In engineering design practice, the designer will switch from one abstraction level to another. Systematic design methods depend upon the designer to make the connections between the steps. The general direction of the design activities however, can be described using the models mentioned.

When a designer starts specifying the details (geometry) of the product, design models do not provide support. The detailed information that becomes available enables analysis of several aspects that influence the realisation of the product, product performance, and costs. Techniques that enable analysis of for instance manufacturing operations needed, available suppliers, and the use of materials during this phase of the design process are limitedly available. Existing design methods focus at the conceptual design phase or enable analysis of a completely specified product.

Design methods

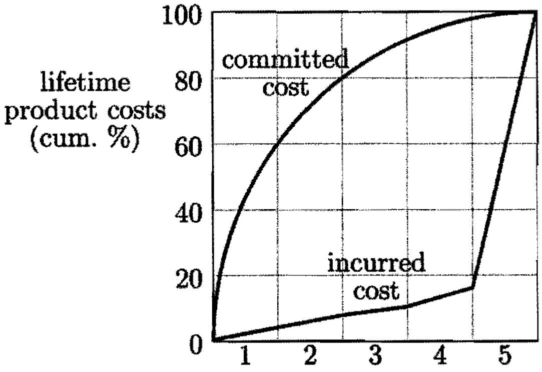
A number of design methods, logical procedures or tools for designing, can be used [Cros89, Rooz95]. They are available for a number of phases in the design process and are creative or rational methods, such as:

- questionnaires, objectives trees, means-end trees (for clarifying objectives)
- interaction matrices, function analysis (for establishing functions)
- performance specification (for setting requirements)
- brainstorming, morphological charts (for generating alternatives)
- checklists, weighted objectives, datum method (for evaluating alternatives)
- value engineering (for improving details)

Some design methods aim at analysing existing products or processes, usually directed at a single performance criterion. Examples are:

- Quality Function Deployment (QFD)
- Design For Assembly (DFA)
- Failure Mode and Effect Analysis (FMEA)
- Taguchi method or Shainin method
- Design for machine dynamics

Quality Function Deployment maintains customer focus [Sull86]. Design For Assembly evaluates products to find opportunities for reducing cost of assembling [Boot82]. Failure Mode and Effect Analysis focuses on reliability of products (design FMEA) or processes (process FMEA) [Ford93, Biro94]. The Taguchi and Shainin method help improve the design of experiments [Byr87, Bhot91]. Design for machine dynamics considers the effects of dynamics on the performance of the overall system [Rank97]. The existence of numerous techniques for design and redesign of products and processes indicates the importance of the design process. This is demonstrated by the impact of the design process on product cost, as shown in Figure 1.7. Whereas most of the costs originate from production, as much as 75% of the cost may be committed in the first phases of the product life [Hubk76, Bake92, Nevi89].



Phases of product life:

1. Conception
2. Design
3. Testing
4. Process planning
5. Production & service

Figure 1.7: Product costs

1.3 Interchangeable parts

Interchangeable parts are a benefit to both users and manufacturers. Instead of replacing an entire system, an individual part is replaced. Manufacturers need interchangeable parts because, for efficiency purposes, they implement manufacturing processes as series of specialised manufacturing tasks (Ford's car assembly line being the most famous example). Such a specialisation of tasks demands for parts that can be used in all copies of a certain product type or product range (or at least classes of copies). This makes interchangeability a prerequisite for manufacturing.

At first, interchangeability was achieved by manufacturing parts after a master that was supposed to be perfect. This system was the result of an army demand for interchangeable parts in weapons to make them less costly (1815-1824). In the middle of the 19^{th} century, parts were first drawn out as separate items, in which the basic dimensions were supplemented by the addition of terms like slide fit, running fit, or press fit. This indicates the actual separation of design and manufacturing activities [Book63]. Since then, parts are manufactured from information given in a product model while interchangeability is to be maintained, so the links between the two are of importance. Besides representation of the ideal, allowable variations of sizes are represented, as manufacturing parts that have the exact nominal dimensions specified is physically impossible. The stochastic behaviour is controlled by tolerances.

As they control the behaviour of a part in an assembly, tolerances are closely linked to the ability of a part to realise a certain function, although the exact relationship is often hard to quantify. Tolerances also determine the manufacturing processes needed for achieving the required accuracy. Any given process has a certain capability where accuracy is concerned. The process capability is defined as the maximum range in size within which a dimension is expected to vary [Bjør78]. Stricter tolerances may result in different manufacturing operations or increase the reject rate of parts. The manufacturing processes used determine manufacturing costs. If multiple processes are available, the manufacturing cost as a function of the tolerance will be a discontinuous line, as depicted in Figure 1.8 [Bjør78]. The tolerances specified are always a compromise between the functions of a product and production cost.

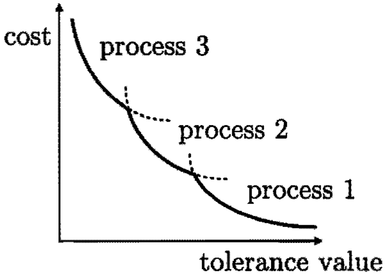


Figure 1.8: Manufacturing costs

1.4 Summary

Designers have to provide product designs in short design cycles and face an increasing amount of information. Design is one of the most important activities in creating a product as it determines most of the cost of a product. To increase efficiency in the creation of products, designers can be supported by means of design methodologies, modelling tools, and manufacturing preparation tools.

Product modelling is dominated by geometrical models of the artifacts. This results in a mismatch in abstraction level and an alienation from the manufacturing phase. Features do not completely solve the problem, as they are domain dependent and pay little attention to assembly activities. Besides that, the design of parts and the design of assemblies are often regarded separate activities and focus on entirely different aspects of product creation. A difficult step in design is the reasoning from function to form, which is supported poorly. Parametric design systems and knowledge based engineering systems provide some support, but they are not frequently used. Design process models provide support in structuring the design process, but should not be applied too strict. Their separation between top down and bottom up is artificial, and they do not provide support for embodiment and detail design. Design methods usually apply to one aspect or phase of a design or provide support for a single performance criterion. Usually, a finished model of a product is needed. Tolerances, which are of importance for functionality and largely determine manufacturing costs, until recently received little attention.

1.5 Objectives and overview

The objective of this research is to improve the creation of assembled products by taking manufacturing aspects into consideration during design. Information about the factory is used to find errors concerning manufacturing and to automate the generation of manufacturing information. This gives the opportunity to reduce costs and lead times and to increase quality, as manufacturability problems may delay physical implementation of the product or make it more expensive. During conceptual design and the early stages of embodiment design, the information available is not sufficient for considering the manufacturing implications of design decisions. Therefore, the emphasis is on supporting the designer in part of the embodiment design phase and the detail design phase. The concepts that are developed should eventually lead to the creation of a design support tool.

As became apparent in this chapter, existing design (for manufacturing) support is mainly concerned with parts manufacturing. This research aims at integrating part design and assembly operations. Besides the part properties, assembly information and tolerance information are incorporated in the resulting product models. The modelling technique used in this research focuses less on geometrical details, uses faces as primitives, and maps design operations onto manufacturing processes instead of using manufacturing features.

In manufacturing, accuracy is important. The adjustment of an assembly however, which especially in high precision machines is of crucial importance to achieve sufficient accuracy, is regarded a separate phase in the product creation process. It is not seen as a part of the assembly process, as it is very complex and the procedures used usually depend on the specific product involved.

To obtain an improved product creation process, the design process, the manufacturing processes, and their industrial environment are studied. A model of the product creation process is used to explain the approach chosen for design support. The next chapter addresses the industrial environment. The product creation process is dealt with in Chapter 3. Chapters 4 and 5 describe the actual product modelling process and its coupling to manufacturing processes. The results are illustrated in Chapter 6 using a case study. Finally, the conclusions and recommendations are provided.

Chapter 2

Industrial systems

The complexity of products and production systems constantly increases. Despite the existence of design models, most designs still depend on an intuitive approach. Interrelationships between products and production systems are often unclear or disregarded. In view of a demand for high quality, short lead times, and cost efficiency, design processes and product models need improving. Structured design processes improve the quality of designs, while suitable product models incorporate the information needed throughout a product creation process. To enable a systematic approach, all activities are placed in the context of the systems theory and the life-cycle concept. This chapter provides a framework that indicates the strong relationship between a product and the production system.

2.1 Systems

Products, machines, and factories are systems, so systems theory is a basic science field in design and manufacturing activities [Hubk73, Hito96]. Although the concept is intuitively clear, a definition of a system is hard to provide. All definitions depend on the discipline involved. To explain the concept it is sufficient to mention the most important characteristics of a system, found in most definitions [Veld72, Aren96]:

- a system is a separated part of the universe defined to serve certain objectives
- a system is a collection of hierarchically ordered interrelated elements
- a system has relations with its environment
- a system exhibits a behaviour

A first attempt to model a work structure by means of the systems approach was made by Merchant [Merc61]. A manufacturing system, is made up of the steps design, programming, control, operation of machines, and fabrication processes, as shown in Figure 2.1. The resulting parts fulfil the design concepts and requirements.

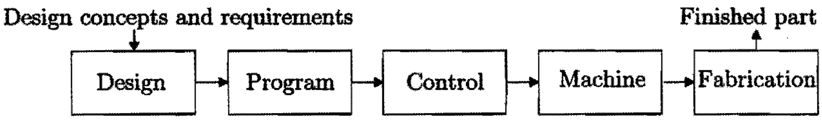


Figure 2.1: Merchant's model of a manufacturing system

This model indicates an important consideration. All steps from design concepts to finished parts should be considered. Some models focus on manufacturing processes, machines, or factories (micro-models). The strong relationships between products and production systems call for a (macro) model comprising the product, the production system, and their mutual relationships. This is called an industrial system [Bran93]. Its primary objective is to generate money by supplying customers with the products required. The model of an industrial system, depicted in Figure 2.2, provides a framework for activities in design and manufacturing.

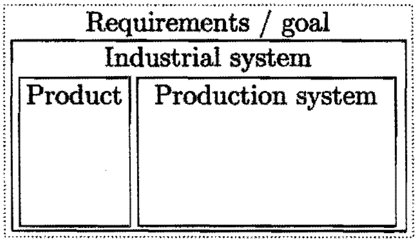


Figure 2.2: Industrial system

2.2 Products

Customers generate a demand, based upon product price, quality, and performance. This demand has characteristics regarding the speed of delivery required, the product variety, and the volume. A good production system reflects the most efficient way to create these products. The technologies used, the lot sizes, the degree of automation, the type of distribution, and the amount of stock are a result of the manufacturers' effort to maximise the added value.

As shown in Figure 2.3, product variety and process variety characterise production systems [Brow96]. Two types of constraints are found: economical constraints and technological constraints. A flexible factory with a great number of processes to manufacture only a few different products is too expensive (economical constraints), while creating a great number of different products with a limited number of processes is physically impossible (technological constraints). Technological constraints only apply to industrial scale production, as artisans create an infinite number of products using a few processes. The various types of production systems, job shop, batch, and mass production, have characteristic products, lot sizes, volumes, and machines.

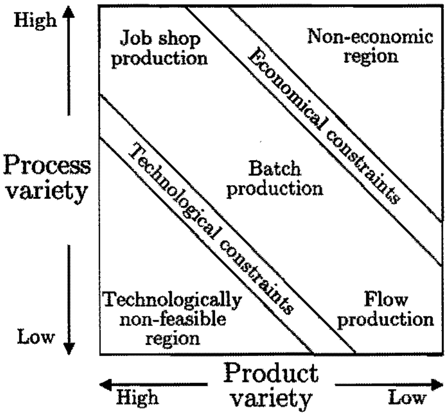


Figure 2.3: Process and product variety

The type of machines used in a production system largely depends on the production volume. Cost performance determines the most suitable alternative. Figure 2.4 shows the relation between unit production cost U and production volume V [Hito96]. Small volumes require general purpose machines, while high volumes justify the use of special purpose machines. Medium volumes can be manufactured using general purpose machines with special tooling or automatic (programmable) machines. Automated generation of control information enables the use of automatic machines in small batch production, as it improves cost performance.

Whereas production volume is determined by the market, lot sizes are optimal values obtained when inventory costs, production costs, and service to customers are considered. Stocks decouple client orders and production processes, controlling the

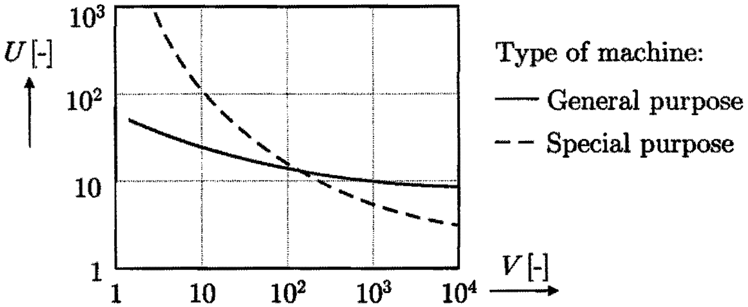


Figure 2.4: Manufacturing alternatives

material flow up to a decoupling point. Production systems with various decoupling points exist, but generally two kinds of production can be identified: stock driven production and customer driven production, depicted in Figure 2.5 (circles depict processes, arrows represent interactions [Rood96]). In customer driven production, manufacturer *Ma* receives orders *o* and delivers products *p* directly from and to customer *Cu*. In stock driven production, manufacturer and customer are decoupled by stock *St*. Both obtain materials *m* from supplier *Su*.

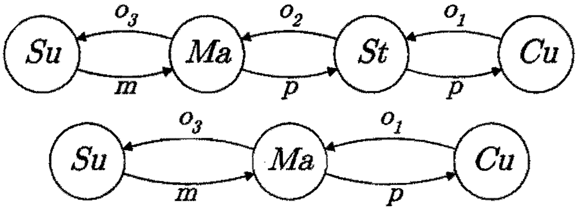


Figure 2.5: Stock driven production vs. customer driven production

The relative shares of the time used for design, manufacturing preparation, and actual manufacturing relate to the type of product and the production volume. If volumes are high and products are only changed incrementally, the time used for design and manufacturing preparation is relatively small compared to the time used in actual manufacturing. In small batch manufacturing of new products, design and manufacturing preparation can establish a major part of the lead time. When a customer order requires a new engineering design to be developed, so-called engineer to order production, design becomes a crucial activity in controlling the lead time. To meet the required lead time, the designer is supposed to find the fastest way to fulfil the functional requirements.

Minimising the number of manufacturability errors in designs reduces lead time. Errors such as parts that cannot be manufactured according to their specifications or parts that do not fit into the assembly influence the lead time as they result into a lot of iterations between the design stage and the manufacturing stage. The specifications are changed and a new set of parts is created, which unnecessarily increases lead time.

The indirect influence of the design should not be underestimated. When manufacturing large batches, effort may be put into finding the best materials, processes, and shape of the parts. Lead times can be shortened by finding faster processes, integrating multiple functions into one part, and by improving the logistic performance of the factory. Large volumes justify the use of special equipment and optimisation of the design and the production system. Design errors usually apply to the first versions of a product and are mostly solved in full-scale production (learning effect). However, because of the batch size, design errors have a tremendous impact.

A reduction in lead time that can be achieved by minimising errors is demonstrated by analysing the origin of the time spent when assembling a special purpose manufacturing machine, as depicted in Figure 2.6 [Phil94]. A similar distribution is found when analysing the manufacture of the parts. A large percentage of the lead time is used for performing activities that do not contribute to the added value, but merely result from manufacturability problems.

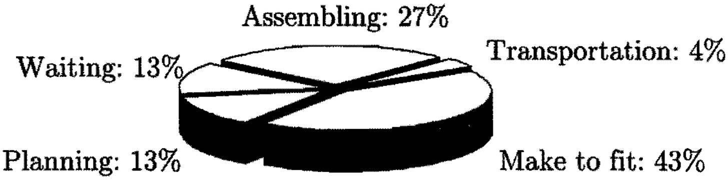


Figure 2.6: Distribution of time spent in assembling

The complexity of products and production systems requires close co-operation between design and manufacturing. Problems occur because of faulty tolerances, tolerances that are hard to realise, insufficient information about the functions of a part, and insufficient documentation on the design intent. Assembly processes act as summation processes in manufacturing. Most of the errors found during assembly operations are not caused by the assembly operations themselves, but are implicated by (accumulated) imperfections of the parts. A survey at Boeing, conducted to determine the causes of assembly problems, showed that problems mainly occur because of non-fitting parts, unanticipated tolerance stack-up, bad co-ordination of tool design and engineering, and interfering parts in the assembly [Shal92].

The increase in lead times caused by design errors demands for an inherently safer way of creating product models. Most profit is gained by focusing on complex products that are created in small batches. An indication of the results was given in [Vrie96]. A product area was targeted consisting of parts with relatively simple geometry, manufactured in small batches using complex, but well formalised manufacturing operations (material removal operations). This product area comprises the mechanical parts that are typically used in special purpose manufacturing machines. Although assembly was not taken into account, a considerable decrease in lead time was found. This was achieved by performing manufacturability analysis during the design process and automatic preparation of manufacturing information.

Careful consideration of requirements on mechanical parts also reduces lead time and costs. When it is prevented that designers create models that are unnecessarily strict (over-design), the manufacturing operations used are less complex. Feedback to a designer on the consequences of design decisions therefore increases quality. Usually, quality is defined as the extent in which the product meets customer specifications. In manufacturing, this is less useful, so high quality is defined as a minimal deviation from the specifications. This definition also causes problems, because the deviations themselves are part of the product model (tolerances). Here, the concept of quality is related to the industrial system used to create it: quality is the ability to perform the required function while using minimal resources (capacity, money, time). For instance, if a part performs a function that demands a tolerance value of 0.1 millimetres, a deviation of 0.099 millimetres means high quality. A deviation of 0.001 millimetres wastes resources. A basic rule in quality assurance, the target level should be as-good-as-necessary, is automatically met [Biro94].

2.3 Production systems

Three subsystems are found in a production system [Rood84, Aren89, Bran93]:

- The primary system (PS) or manufacturing system involves the flow of material. The material flow is all that is subject to a transformation.
- The secondary system (SS) or control system is associated with the flow of information that controls the material flow.
- The tertiary system (TS) or economical system incorporates the flow of money, compensating the flow of material. This is needed to preserve the system.

This model of an industrial system is depicted in Figure 2.7. Seeing both the product and the production system as part of an industrial system, emphasises their close interrelationship. The information that is exchanged between the products and the production system is controlled by process planning and production planning.

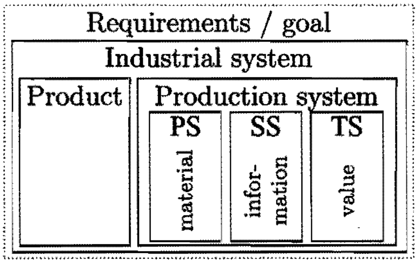


Figure 2.7: Subsystems in an industrial system

Basic operations perform the transformations on the material flow: processing operations and supporting operations [Smit92]. Processing operations, like shape changing, assembly, and inspection, transform raw material into parts or products that meet the design specifications. Product descriptions are converted into working instructions for the control system. The activity that deals with this conversion is known as process planning, depicted in Figure 2.8 [ElMa93]. Process planning generates part of the information flow, such as machining methods, tools, set-ups, machining sequence, process conditions, tool paths, and operator instructions. The information is generated by modifying standard plans of similar products (retrieval or variant approach) or by using decision making logic and algorithms (generative approach). In both cases, it is not a one-way activity. Information about the available resources is needed to perform correct process planning, such as process capabilities, machine models, and information about tools and materials.

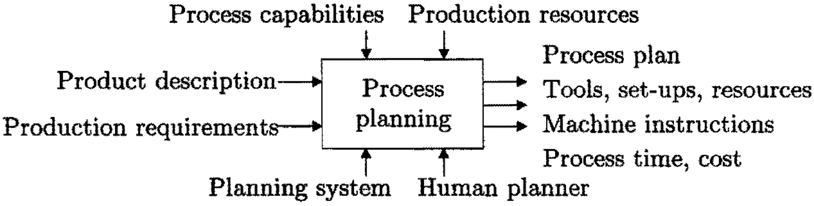


Figure 2.8: Process planning

Process planning information is not sufficient to perform activities such as scheduling or controlling the work load. Most supporting operations, such as materials handling, transportation, and storing are not covered by process planning either. Supporting operations and batch level information are controlled by production planning, which generates the remaining information needed to control a manufacturing system. Although process planning and production planning use the same resources, this research focuses on the relations between design and processing operations only.

2.4 Life-cycle

A system is defined to serve certain objectives. As objectives change, new systems emerge and old systems disappear. To describe the status of a system in time, the life-cycle concept is used. This concept is illustrated in Figure 2.9 and holds for both products and production systems [Rood96].

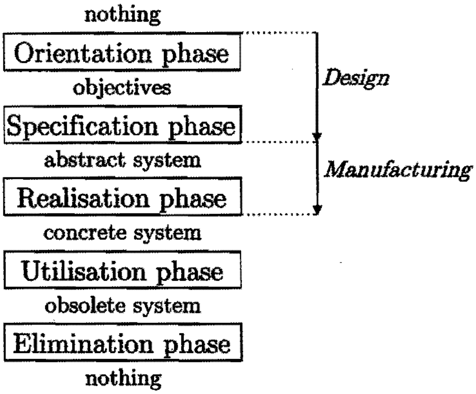


Figure 2.9: System life-cycle

In the orientation phase, the objectives (initiated by the problem to be solved) are defined. After this phase, one is aware of a certain need. The functions that have to be performed to satisfy this need are defined in the specification phase. It is also defined how these functions are performed and the resources they require. The definitions are captured in an abstract system or model. In the realisation phase the abstract system is built and tested, which results in a concrete system; something that exists in physical reality. It is used in the utilisation phase to get a return on the investments that were made in the previous phases. If the objectives change or the concrete system does not meet its specifications anymore, it has become obsolete. It is eliminated in the elimination phase, resulting in nothing (an optimum reached only if the system was created to allow for complete elimination).

The specification phase

The specification phase deals with the functions of a system. Objectives are usually stated in terms of the functions a system has to perform. Methodical design authors use a transformational view of function, as expressed in the VDI definition [VDI87]:

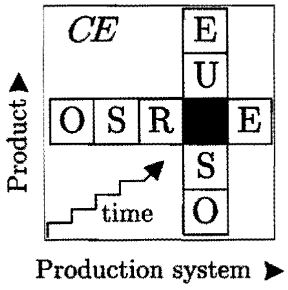
> *A function is defined as a relationship between the input, the output flows, and the state variables of a system, independent of a particular solution. The input and output quantities may be energy, information, or material.*

Functional modelling structures problems into solvable sub-problems by defining sub-functions. A designer is supposed to define these sub-functions and create a description of a product that is able to match them. The output of the specification phase is a model that specifies part geometry, materials, and assemblies. This output is used in the realisation phase to create a physical product by means of manufacturing processes. Product form determines the ability to meet the specifications.

Concurrent engineering

In 1988, the Institute for Defense Analysis provided a commonly accepted definition of concurrent engineering, stating that concurrent engineering is a systematic approach to the integrated concurrent design of products and their related processes, including manufacture and support. The objective of concurrent engineering is to help overcome the problems of increasing complexity and shorter design cycles. It promotes the interchange of information between disciplines, such as marketing, engineering, service, and manufacturing.

When considering a product and a production system simultaneously, this demonstrates the importance of concurrent engineering. In its utilisation phase, a production system creates a product from a product model; the realisation phase of the product. Design decisions should not be made without considering the consequences for the production system. In the course of time, designers alternately decide on properties regarding the product and the production system. Thus, design is depicted as moving across a concurrent engineering plane (CE) described by a product axis and a production system axis, as shown in Figure 2.10.



Phases of the life-cycle:
- Orientation (O)
- Specification (S)
- Realisation (R)
- Utilisation (U)
- Elimination (E)

Figure 2.10: Concurrent engineering

Competitive strategies

Until now, a certain product and production system were observed. From a competitive strategy point of view however, one has to bear in mind that companies usually manufacture a gamut of products. One can also define a life-cycle that refers to rate of growth of product sales [Port80]. Products pass through the phases introduction (I), growth (G), maturity (M), and decline (D), as depicted in Figure 2.11. When a product is introduced, buyers do not immediately react, so the curve is relatively flat. When a product gets successful, sales grow rapidly until all potential buyers are reached. Then the growth stops and the sales level off. When new products appear that substitute a certain product, the sales will go down. Besides the S-shaped curve, other patterns are found, such as style and fad life-cycles [Kotl88].

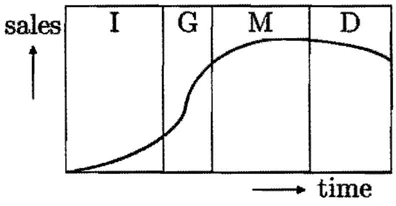


Figure 2.11: Product life-cycle

The various stages in product life-cycles result into a shift in the way a company competes. This also influences the manufacturing of the product. Above all, it is wise to have a range of products that are in different phases of the life-cycle. This provides a balanced work load and cash flow in the course of time.

2.5 Summary

In this chapter, systems theory and life-cycle concepts were used to put the design and manufacturing activities into perspective. The relations between a product and its production system have been pointed out.

Product and process variety determine the type of production system. Production systems have characteristics, such as the type of machines used, the lot size, characteristic products, and volumes.

An important issue is the influence of the design process on the lead time. Preventing manufacturability errors in the design process reduces lead times, as these errors result in activities that do not contribute to the added value. These activities concern redesign and manufacturing of parts that do not fit into the assembly or waiting for parts that could not be manufactured or supplied. A designer should receive feedback on design decisions to improve product models. A product creation process that prevents manufacturability errors to appear in designs reduces lead times and increases added value. Product models that are free of errors are also needed for automated process planning. Such a product creation process embodies design, process planning, manufacturing, and some form of feedback to the designer. Thus, it supports the principles of concurrent engineering.

The product creation process is studied as far as design and manufacturing processing operations (material removal, material adding, forming, assembly, measuring) are concerned. Aspects like the flow of values, factory capacity, maintenance, reliability, and recycling are not taken into account. The result of the study is a model indicating the processes involved and their mutual relations, presented in the next chapter. This model is used to explain the approach chosen.

Chapter 3

Creating products

In this chapter, a model of the product creation process is introduced to explain the approach that is followed to improve lead times and cost efficiency. This also indicates processes that can be performed by a design support system. In graphical representations, circles depict processes, while arrows represent interactions [Rood96]. When a suitable model is found, the processes used in that model are explained in detail.

3.1 The product creator

Industrial systems incorporate flows of material, information, and values. If the values flow is omitted, this can be modelled as depicted in Figure 3.1. Customer C is supplied with products p that satisfy functional requirements f. A product creator PC transforms the requirements into products (after [Delb89]). The product creator places orders o to obtain materials m from supplier S.

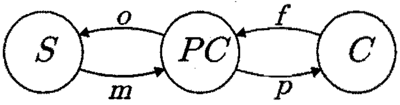


Figure 3.1: Industrial system

By examining this model in more detail, the approach followed can be explained. A product creator can be imagined that consists of four co-operating processes: designer D, process planner P, realisator R, and evaluator E, as depicted in Figure 3.2. The model represents the processes present in a product creation process. These processes are performed by —but not unambiguously coupled to— persons, departments, factories, or CAD systems. At first, only the processes and interactions themselves are of concern. When a more suitable model is found, the person, system, or department performing a certain process is pointed out.

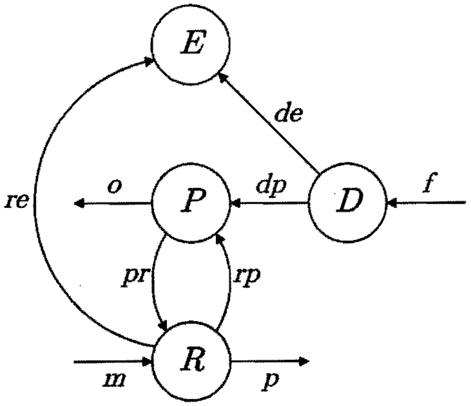


Figure 3.2: The product creator process

Designer D creates product model dp based on functional requirements f. Although a product model contains various information, the main concern here is the information used in manufacturing. Other information created by the design process, such as specific software or dynamic models, is disregarded. For manufacturing planning, dp is usually in the form of drawings or geometrical models, which are sent to process planner P. The process planner interprets the model and describes a collection of processing operations to transform materials into parts and assemble these parts into a product (manufacturing job pr). Materials are ordered from a supplier (orders o). Information about available resources rp is needed from the process that performs manufacturing operations, realisator R. The realisator receives a manufacturing job and performs the operations specified, resulting into product p.

Product p should meet requirements f made at the beginning of the design process. To make sure, evaluator E is provided with data re from the product. This data is compared with design requirements de provided by the designer. If the requirements are not met, the processes are performed again until the evaluator proves the realised product to be correct. In this research the focus is on manufacturing aspects, so evaluator E is reduced to a manufacturability evaluator. Consequently, design requirements de are geometry related and product data re mainly concerns realised form. If the designer process creates a product model containing errors like incorrect dynamic or thermo-mechanical behaviour or errors caused by a lack of knowledge about the processes the product performs, this is not detected because the realised form is considered correct and manufacturing operations are found.

Feedback on the realised form is common practice. However, in this model manufacturability problems can occur when product model *dp* is sent to the process planner. If the designer did not observe manufacturability, the process planner may be unable to find suitable processing operations. A manufacturing job cannot be created, so the product model has to be modified. Most companies provide mechanisms to avoid such problems. In other words, the product creator sketched lacks several feedback channels. A designer process without feedback and changes in behaviour does not exist. At least, feedback from process planning, realisation, or evaluation processes comes in the form of so-called *noise from the factory floor*. More formal feedback is used in the form of exception reports or corrected drawings that are returned to the designer. However, this feedback is often late and incomplete and therefore it takes a lot of time and effort to modify the product model.

An approach that aims at finding manufacturability problems before the product model is sent to the process planner should systematically provide the designer with knowledge on manufacturability of a product model. The designer should use information about the processing operations used to create a product (re_p) as well as the realised form of the product (re_f). This form of feedback, specifically used in a design tool, is depicted in Figure 3.3 [Delb89]. The information is fed back to the designer in the form of manufacturing restrictions *ed*. If the designer observes the restrictions, this reduces problems in finding suitable processes for manufacturing parts or in assembling these parts.

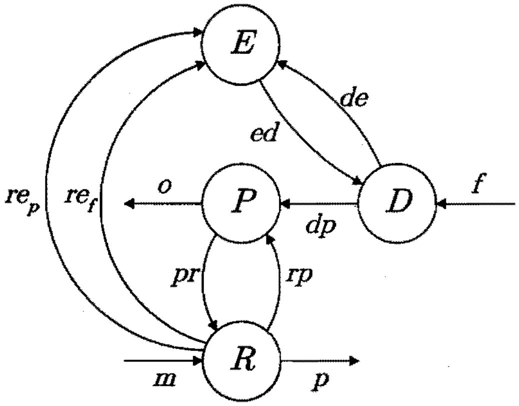


Figure 3.3: Manufacturing restrictions

In evaluation, a product specification is considered manufacturable if at least one series of processing operations is found that can be used to create the product according to the requirements. It refers to the availability of processes, tools, and materials, as well as to the capability of the processes. This includes the capabilities of subcontractors and the properties of parts from outside suppliers. As product costs are also part of the requirements, designs requiring operations that are available but are extremely expensive comprise a manufacturability problem.

The product creator depicted in Figure 3.3 indicates the general approach used to decrease the number of errors in the product model and thus reduce lead time. Manufacturability is maintained by using rules the designer should observe and by evaluating the resulting products. The model can be improved however, as it does not illustrate the approach in sufficient detail.

Making manufacturing restrictions available to the designer does not guarantee manufacturability of the product models. The rules should be used to automatically evaluate the product model during design to ensure that the design process inherently produces manufacturable product models. Evaluation tools that require a final design generally increase design time. If evaluations lead to corrections, it usually takes a lot of effort to perform the corrections if the design is in the final stage. As the designer is the one possessing knowledge about the design intent, immediate feedback to the designer is (time) critical. Besides that, information on the current product model, such as material, shape, and so on, is needed to perform a correct manufacturability check.

The model is changed to accommodate for these drawbacks. The new model indicates the fact that manufacturability evaluation is performed after adding a single data element instead of waiting for the designer to finish a part or sub-assembly. The processes work with partial specifications, that is, incomplete product models. In the graphical representations, the processes and data used in the continuous checking are indicated with an asterisk ($*$). The product creator is split into two separate parts. The first part, depicted in Figure 3.4a, no longer contains a realisator process.

Designer D^* receives functional requirements f on the product that has to be created. This design process D^* is able to generate intermediate product models dp^*. Evaluation of manufacturability can only be performed if manufacturing information is generated. Process planner P^* performs process planning on incomplete product models. It creates collections of manufacturing operations pe^* that are necessary to meet the specifications in the product model. Manufacturability evaluator E^* checks whether these manufacturing operations can be performed using the resources available (process capability, tools). Evaluation result ed^* is immediately forwarded to the designer. Product model dp^* is a model composed of separate design decisions made by the designer. Each time a design decision is added to product model dp^*,

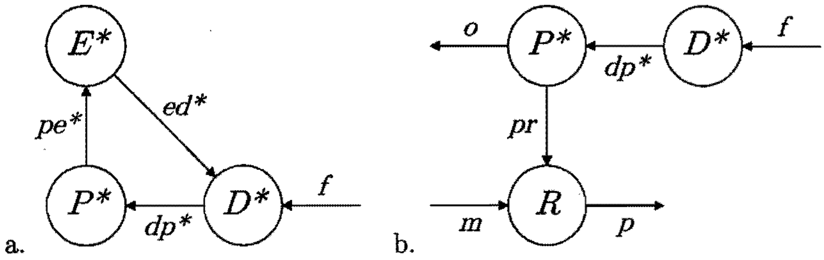


Figure 3.4: Improved product creator

process planner P^* adds a corresponding set of processing operations to pe^*. If a design decision causes manufacturability problems, the designer can change this decision, or one of the previous decisions, to solve the problem.

Design decisions are added until functional requirements f are met. The resulting dp^* is a final product model. As realisator R has not changed, its input (manufacturing job pr and material m) and output (product p) remain unchanged. The second part of the product creator creates the product from the final product model, as shown in Figure 3.4b. Because the product model is composed of elements that were individually checked for manufacturability, no problems occur when generating a process plan (provided each check takes all design steps into account, that is, after each step the entire product model is evaluated). Therefore, it is called the manufacturing of a *right first time design*. The absence of iterations between the design process and manufacturing process reduces lead time. The model depicted in in Figure 3.4a and b indicates an integration of design and process planning activities. Each data element or design decision is coupled to process planning information. As a result, the boundary between design and process planning becomes less strict.

To generate manufacturability rules concerning realisator R, evaluator E^* is kept up-to-date with information about the processing operations and the resulting products. It is also needed to supply E^* with information about availability of resources, such as tools, and information on other resources (sub-contractors, suppliers). Process planner P^* decides what operations to use to meet the specifications, so it is supplied with the information generated by the realisator (processing operations and resulting products). To generate a manufacturing job, machine data is needed, such as machine configurations. Updating the evaluation process and the process planning process with information generated during manufacturing creates a learning system. The evaluator reflects the actual factory capability and process planning optimally uses the available processes. Updates are performed using data that is already present (or should be present) in current production systems. Process capability, tooling information, and machine configurations for example are already used, although not systematically and mainly based on human experience. Theoretically, automated process planning and incorporation of factory capability enables the design of products while keeping current utilisation of machines in mind. However, the different spaces of time used in design and manufacturing usually prevent this.

Information about parts or processes from outside suppliers is not always available or constantly changing. The designer will have to add this information to the design in some cases. For instance, when using parts from outside suppliers, the tolerances of the surfaces contacting other parts in an assembly are added manually (if available).

The product creator model indicates the approach for improving the product creation process. At this point, an indication can also be given on the persons, systems, or departments performing a certain process. Processes P^* and E^* can be performed automatically by a computer system, while process R consist of a production system. Designer process D^* is more complicated. Part of this process is performed by a human designer (or group of designers). The other part consists of a design support system that is used to create and store the product model. The next paragraphs elaborate on the nature of the various processes in the product creator.

3.2 The designer

The product creator treats design as an evolving process, involving a sequence of design decisions. These decisions are incremental steps towards a product model fulfilling functional requirements. Evolutionary models of design are mainly found in literature concerning design history or design rationale. It is argued that design support systems provide an excellent means of representing and communicating final design specifications, but lack the ability to recount the process that leads to the final product [Chen90, Aasl93]. Developers of design history tools try to establish techniques for formally representing design knowledge.

The product creator requires formalised steps or decisions in the design process. A major drawback of the models used in design history research is their high level of abstraction. The history of major design decisions is supported instead of detailed functionality, geometrical models, and manufacturing information. The tools that are a result of these models capture the process but are not active design tools. It can be concluded from the design history process models however, that the design process model and the model of the artifact to be designed are closely linked. Therefore, an approach that is based on a process representation is likely to succeed.

The Evolutionary Design Process Model [Tomi89, Xue92] or General Design Theory

[Veer89], depicted in Figure 3.5, supports active design. The status of a design representation, the metamodel, is changed (to a current incomplete description of the artifact M_i). A designer observes the current status of the metamodel and decides what to do next. A metamodel is decomposed into several aspect models (m^i_j), each focusing on particular properties and attributes. The aspect models are evaluated by the designer to find out if a step is successful (evaluation e_j).

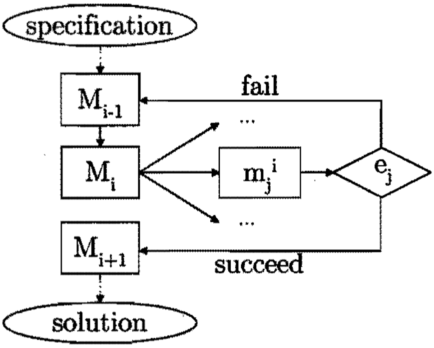


Figure 3.5: Evolutionary design process model

The basic design cycle model focuses on design activities rather than the representation of the solution [Rooz95]. It states that design is a sequence of empirical cycles, in which the knowledge of the problem as well as the solution increases spirally. To solve a problem, one must go through the basic design cycle at least once. It consists of five steps: analysis, synthesis, simulation, evaluation, and decision.

- In the analysis phase, a designer forms an idea of the problem and formulates criteria that a solution should meet. At first, this idea will be broad. In later iterations, it will be more accurate and complete.
- A provisional design proposal is generated in the synthesis step. Separate ideas are combined into an integral solution (in which creativity plays a part).
- Preceding the actual use and manufacturing, the properties and behaviour are simulated, leading to expectations about the actual properties.
- In the evaluation step, the value or quality of the provisional design is established by comparing expectations and criteria.
- After these four steps, a decision follows. It is to be decided if the design is a final design or if another iteration should be started.

In [Bran93] it is stated that simulation (mental or formal) is part of evaluation, so a four step cycle results (the steps analysis, synthesis, evaluation, and decision are equal to the ones described above). Both design cycles are depicted in Figure 3.6.

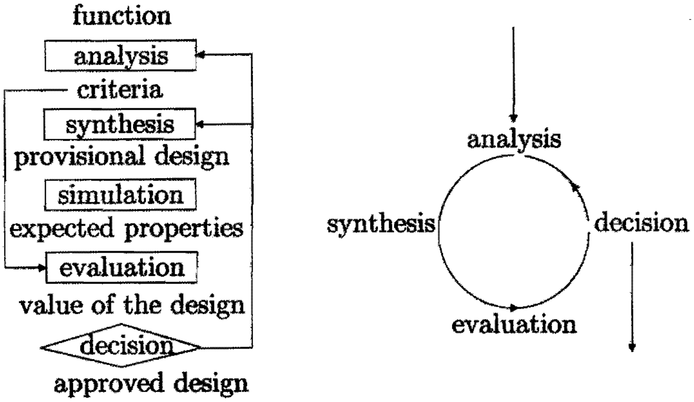


Figure 3.6: The basic design cycle

The models described above follow an evolutionary approach, but do not strictly separate the evolution of the product model and the mental processes of the designers. The evolution of a product model, especially concerning the geometric details, can be described using for instance the Evolutionary Design Process Model. The basic design cycle approximates the less formal part of the design process. It is suited for describing a design process that includes analysis and simulation of geometry, static behaviour, dynamic behaviour, and thermo-mechanic behaviour [Sche98], but not for representing the evolution of the product model.

Here, designer process D^* is separated into a formal and a non-formal part. The formal part of D^*, establishing and maintaining a description of an artifact, is modelled as an evolving process. The description is specified by a (human) designer responsible for the creative, non-formal, part of D^*. The non-formal part is concerned with the generation of concepts and ideas and the arrangement of functions into physical blocks. The formal part of the designer process is represented in a design support system and constitutes the connection to the process planner and the evaluator. For now, the non-formal part is too difficult to be incorporated in a design support system and therefore is left to the human designer (see Appendix B).

The boundary between the parts consists of the instant in which a designer describes and records the result of the mental process; a detailed, unambiguous description of the artifact (geometrical design). Here, this instant will be different from the one in conventional design. Instead of merely drawing the result of a number of design steps, designers need to formally specify the steps to enable systematic evaluation. Design support tools guide a designer through the correct procedure and are not created to support conventional design procedures.

An important aspect of the basic design cycle is the use of the steps simulation and evaluation. These steps were also used in the product creator model. Based upon the description of an intermediate design, the manufacturing operations needed are evaluated. This can be regarded a simulation of the manufacturing phase. The expected operations are evaluated and based upon the result, a decision is made. The differences are the strict separation of mental and formal processes (and thus mental simulation and actual experiments) and the different time-scale.

The basic design process model

The concept of states and state transitions is used in the formal part of design and manufacturing to create a basic representation and terminology. The design process is modelled as a series of time-dependent actions that transform the information through a series of states, as depicted in Figure 3.7. The states define a design at each point in its development by representing space, time, and properties [Onos89, Salu91]. Basic operations in representing a product design are:

- generating a state by specifying space, time and property
- modifying a state by changing its space, time, and/or property
- deleting a state (states caused by it are deleted as well)
- composition of a state, in which two states are merged into another state
- decomposition of a state into sub-states

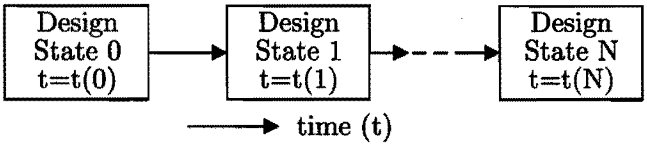


Figure 3.7: State transition model of the design process

The latter operations refer to the modelling of assembly operations; assemblies are sets of sub-states. The concept of states and transitions is applied to geometrical design and to manufacturing. This results into design states and design transformations to represent geometrical design. Manufacturability fits into the concept by identifying manufacturable transformations; transformations that passed evaluation.

The product model

The design history is used as a product model by storing the intermediate results produced when designing. If a designer changes design states, the new collection of states and transformations is kept, so a product model is an idealised recording of a design process. Product geometry is inferred from the states and transformations.

The advantage of this model lies in the availability of the separate transformations. Transformations can be evaluated, altered, or grouped (for process planning). New information that is added to the design is likely to influence previous design transformations. Therefore, all design information is needed for evaluation and to infer the description of the product. Product models containing only the result of design decisions lack information that is crucial for mapping design operations onto manufacturing operations.

In geometrical design, one needs to represent functional aspects of geometry. Essential parts of design constraints are more or less related to geometrical constraints. A designer needs the means to specify geometry in a way that reflects the function of the product, which requires a formal representation of a relation between function and form. The transformational view that was presented in Chapter 2 is not sufficient. A description that reflects the relation between function and form needs the concept of relations [Ullm93].

> *What is called function is realised by both transformations and the cause of the transformations. Function and behaviour during design are developed through specifying and identifying changes in the attributes of an object and the relationships between objects. The relations embody functionality.*

Relations are geometrical relations, tolerance relations, and assembly relations. Geometrical relations enable a distinction between the essential geometry of a part (such as mating surfaces) and the inessential geometry. Most products are at least an assembly of modest complexity, so assembly relations are needed. As functionality is not captured by nominal geometry alone, tolerance relations are required. The use of relations instead of drawing the resulting artifact shifts the border between the formal and the non-formal part of the designer. The formal domain is extended because creating a description of the artifact is possible at a point that was formerly in the mind of the designer.

3.3 The process planner

The process planner maps product information upon manufacturing operations for two reasons. It generates the information needed for simulation of the manufacturing operations (evaluation) and creates manufacturing jobs (realisation).

Process planning for evaluation

For evaluation purposes, a relatively rough process plan is sufficient. A design process step, that is, a collection of geometrical relations, tolerances, and assembly relations, generally has a manufacturing counterpart. It is to be determined what processes are suited to render the results required. Various classifications of processes are provided, such as by Kienzle and in DIN 8580. These classifications only consider manufacturing processes. Smit proposed basic operations based upon a production system view instead of a manufacturing process view [Smit92]. The terminology used in this classification corresponds with the one used in this research, so a selection of basic operations is used here: shape, transform, and assemble operations.

- shape operations create discrete products from bulk goods (extrusion or casting, using materials like granulated plastics or ore)
- transform operations are add operations, remove operations, or form operations, in which material is added to or removed from a discrete product, or its properties are changed (turning, milling, bending)
- assemble operations put together discrete products to create a new one

A structure called a recipe contains the operations used to make a product. For manufacturability evaluation recipes lack detail, so the concept of a micro-process plan is proposed. A micro-process plan describes the manufacturing processes used in a basic operation, including the machines and tools needed [Ferr90, Groo93a]. The processes are found by comparing geometry, materials, and tolerances with the capability of the processes available. For example, if a designer specifies an accurate hole, it is manufactured using the basic operation remove, while the micro-process plan contains the processes centre-drilling, drilling, and reaming.

The type of design operation and the type of basic manufacturing operation do not have to match. A design operation that for instance adds a protrusion to a discrete product, is not necessarily accompanied by a manufacturing operation of type add. Design operations and manufacturing operations are decoupled (as opposed to manufacturing features), so it is possible to create a micro-process plan containing material removal operations for creating a protrusion in a design. Such mappings require several extra steps, performed by the process planner, which are extensively dealt with by De Vries [Vrie96].

It is also possible to map a design operation upon several basic manufacturing operations. It is better however, to aim at a single basic operation for each design operation. This limitation is caused by the differences in process planning for add, remove, form, and assemble operations. A basic operation needs procedures for generating process plans. For separate basic operations, knowledge bases can be created. If combinations are allowed, the number and complexity of procedures will increase exponentially. Consequently, complex design operations are split up. For instance, when connecting two parts by a peg-in-hole connection, the designer provides design operations for changing part geometry (create a peg and a hole) as well as for positioning the parts relative to each other. These operations are mapped upon transform operations and an assemble operation respectively.

Process planning for realisation

When a product design is finished, a collection of micro-process plans exists that describes the manufacturing operations needed to create the product. This collection however, is not a manufacturing job. The major differences lie in the existence of set-ups and precedence relations. A set-up is a position and orientation of the product being manufactured (relative to a machine). The manufacturing operations are distributed over the various set-ups. This distribution depends on precedence relations between the operations and tolerance relations. In a number of cases, a manufacturing operation is located at a position that can only be reached after, or before, another operation has been performed. Furthermore, if manufacturing operations concern faces that have a tolerance relation, these operations are kept within the same set-up (if possible).

When creating a manufacturing job, choices have to be made concerning the production system. The machines are chosen, along with tools, fixtures, and materials. These choices depend upon achievable machining accuracy, availability of resources, costs, and processing time. A number of optimisation criteria can be considered, such as minimising the number of set-ups and tool changes, creating stable subassemblies, and performing operations in parallel. Generally, this results into large changes in the sequence of the manufacturing operations. Together with precedence relations and tolerances, this makes the planning of sequences, set-ups, and fixtures a complex task [Boer90, Delc92, Groo93a].

The resulting manufacturing job contains tool paths (numerical control code for part manufacturing operations, inspection operations, and assembly operations), a list of tools, grippers, and fixtures, a list of parts from outside suppliers (parts contracted out, fasteners, bearings, coatings, and lubricants), process conditions, and operator instructions. Creating a manufacturing job is not a crucial activity for the product creator process. As manufacturability was already evaluated, the creation of detailed information could also be performed externally. Specific (software) tools exist for this task, such as assembly planning tools and process planning tools for milling or turning. Most tools are supplied as extensions to CAD tools or as a separate tools with a specific interface. Transferring the product model (as specified in the product creator) however, can result in the loss of valuable information, especially concerning part relations and tolerances. Besides that, the relation between process planning and factory capability is not always observed in these tools.

Capability

Knowledge on manufacturability is not generic knowledge but depends on the industrial system involved. The resources of the industrial system are documented to create a knowledge base and procedures decide what manufacturing processes to use. This suggests the use of statistical and data analysis techniques, such as the manufacturing process capability C_p [Holl95, Crev96, Swif97]. This number compares what is required for a product to function properly (specification limits) to what is (economically) possible for a process to deliver. It is determined by the ratio of the tolerance latitude (upper minus lower specification limit: $USL - LSL$), and the measured variability of the process output. Standard deviation σ characterises the process output (normal distribution). A guideline that is often used is the 3σ quality standard. A process should have a σ that enables the specification limits to stay within the -3σ to $+3\sigma$ range: $C_p = 1$. The fact that the process output average value $\bar{y}$ may not be on target T is not taken into account. This is corrected by using the C_{pk} value. The C_p and C_{pk} value are calculated as:

$$\begin{aligned} C_p &= \frac{USL - LSL}{6\sigma} \\ C_{pk} &= C_p(1 - k) \text{ in which } k = \frac{2 \mid \bar{y} - T \mid}{USL - LSL} \end{aligned}$$

The manufacturing process capability provides information on the processes in a factory. However, it is a process control tool that uses specification limits. This makes it hard to use in a design environment, as the influence of the product geometry is not taken into account. International standards dealing with tolerances and fits provide so-called international tolerance grades or quality numbers IT (ISO2861, ANSI B4.2, DIN7151, NEN2802). These numbers are based on product geometry and are available for cylindrical faces with characteristic length L:

$$i(L) = 10^{-3}(0.45L^{1/3} + 10^{-3}L)$$

Associated quality numbers are: 10i=IT6, 25i=IT8, 100i=IT11, and 1000i=IT16, which are equidistant lines on a logarithmic scale. Quality numbers express manufacturing complexity and are coupled to manufacturing processes. A lower number represents a higher manufacturing complexity. De Vries extended the definition of quality numbers to applications other than fits [Vrie92], based upon empirical data. Figure 3.8 depicts a graphical representation of factory capability. Each field represents a series of manufacturing processes. The width and position of the fields is different for various factories. The fields (combined with surface finish data) accurately reflect the capability of the processes, related to tolerances and actual product geometry. Therefore, it is more useful than a process capability number or general guidelines on the selection of processes.

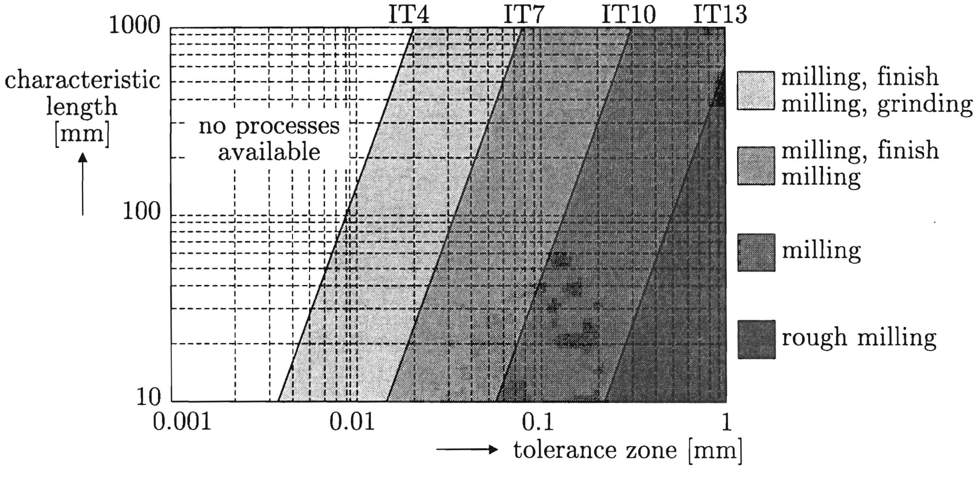


Figure 3.8: Factory capability

The quality number representation is available for material removal processes and partly for forming. It is also needed to reflect the capability of other manufacturing processes, especially assembly. This enables a selection between various forms of assembly equipment, such as manual assembly workstations, robots, selectively compliant robot arms (SCARA robots), flexible assembly cells, and so on. Besides that, feeding equipment, transport systems, and grippers can be considered. For joining techniques such as welding and gluing, parameters can be found. In assembly, the uncertainty in the position of the mating parts is the most important. It is expressed in the quality numbers of the mating faces and the accuracy of the assembly equipment. Other part characteristics that are of importance are expressed in the classifications made by Boothroyd, such as weight, stiffness, symmetry, and nesting properties [Boot82].

Clearly, assembly process selection is complex, that is, it depends on multiple parameters. Instead of a two-dimensional representation as depicted in Figure 3.8, assembly capability concerns more than two dimensions: accuracy (uncertainty), geometry, weight, and compliance. Besides that, two other aspects are of concern. First, virtually no data is available on common equipment for assembly. Although specific assembly manipulators are available, the effort aimed at finding generally applicable equipment is relatively small. An exception is the development of robot applications, although robots are not always optimised for assembly tasks. Second, in assembly, the relation between manufacturing processes and the equipment used is not very strict. Various types of equipment are able to perform an operation. Selection of assembly equipment is largely influenced by economic considerations, making the lot size one of the most important product characteristics. Because the products targeted here are manufactured in small batches, the range in equipment is from manual workstations to flexible assembly cells or robots.

Therefore, the focus is on detecting assembly problems determined by the product design, that is, impossible situations. The selection of equipment is of less concern. It will be limited to selecting a general type, such as a manual assembly workstation or a robot. As a consequence, tolerances, assembly directions, degrees of freedom, assembly sequence, repeatability, and reachability are the main concerns. Quality numbers are used as a tool for reasoning with tolerances.

3.4 The evaluator

The evaluator is provided with results from the process planner (a collection of micro-process plans). The result of the manufacturability evaluations is sent to the designer. The evaluator may encounter five different types of problems:

1. A design operation was specified that has no manufacturing counterpart (the micro-process plan is empty). This problem is caused by product form, materials, roughness values, and geometric tolerances. Especially when using exceptionally strict tolerance values, no manufacturing processes are found (refer to the left most field of Figure 3.8). Another example of this type of problem is parts in an assembly that do not fit, no matter what assembly process is used. This is caused by unanticipated tolerance stack-up or errors while modelling.
2. An operation cannot be manufactured using some specific technologies. A process plan is created, but the technology is not available in the factory (or at outside suppliers). Generally, the process planner will not be able to freely choose any manufacturing process. If a process is not available a manufacturability error is reported. In a number of cases, the processes available are selected at the start of the product creation process, that is, a design is created with some specific manufacturing technologies in mind.
3. An operation cannot be made by a specific piece of equipment. The manufacturing process is available, but there are problems finding a suitable machine or tool. This is related to machine models and databases with tools. The problems are caused by the limited number of machine configurations or tool configurations (machine reach, tool radius or length, grippers, or fixtures).
4. An operation was specified that in itself can be made by available processes and equipment, but the location cannot be reached. Reachability relates mainly to product and tool geometry. In a process plan, approach directions are specified for each manufacturing process. Therefore, it can be checked whether a collision-free path exists for reaching the desired location. For instance, inserting a part at a location that is blocked by another part comprises such a problem. This problem may be resolved by operations that follow.
5. Even if no immediate manufacturability problem occurs, the designer can be warned if an operation requires processes that are very expensive, very time-consuming, need an outside supplier or special tools, or uses parts that are new to the company. If the designer has no special reasons for demanding such a process, the specifications are preferably changed.

The evaluator requires a complete knowledge base and a suitable product model. As stated before, the knowledge is already present in an industrial system. Process capabilities are established and a database of machine models and tools is created. For reachability checking, the volumes described by a tool when performing an operation are determined (for instance for milling tools, grippers, tools used in forming or inspection, welding torches, or wrenches). In a product model, an important aspect concerns a suitable representation of the tolerance information.

3.5 Rapid prototyping and manufacturing

Rapid prototyping and manufacturing generates objects directly from a CAD database, mainly using layer-additive processes, as explained in Appendix A. Currently, the reasonable time and cost of obtaining the models encourage experimentation. This results into savings by minimising the number of mistakes in design, analogous to the design approach mentioned before. Therefore, the position of rapid prototyping and manufacturing in the product creator is examined. Appendix A provides a model of a product creator including rapid prototyping and manufacturing.

Applicability depends largely on the material properties and accuracy of the models. This indicates the major flaw of the rapid prototyping technique. The materials and processes are different from the ones that are used to manufacture the actual product. Evaluation of a design is based on realised form of the prototype. Information concerning the processing operations used is not useful. Besides that, it is impossible to create a prototype after each detail decision, making the feedback loop still too long. Therefore, it cannot be used instead of the model presented in Figure 3.4, but only as a complement. Instead of bypassing the realisator completely, rapid prototyping can be used to periodically create prototypes of products. Although these prototypes should generally approach the actual product as close as possible, creating a rapid prototype provides a means of communication. In the mean time, developments in rapid prototyping provide materials and levels of accuracy that expand the applicability of prototypes. On the other hand, automated process planning can be performed using machines that are immediately available in factories, which enables rapid prototyping using conventional manufacturing techniques [Vrie96].

3.6 Summary

A product creator model was presented to explain the approach that is followed to improve lead times and cost efficiency. A product creation process was proposed, that evaluates the intermediate product model during the design process. The approach presented integrates design and process planning to provide feedback after each separate design decision or data element that is added. The product creator contains a designer process, a process planner process, an evaluator process, and a realisator process. These processes were further explained.

The designer process proves to be a complex process and is explicitly separated into two parts. The formal part comprises an evolutionary process, consisting of separate steps or cycles that include simulation and evaluation. In each step, a number of relations is specified to represent product functionality. The non-formal part of the design process is left to the human designer. The process planner deals with the mapping of design operations onto manufacturing operations. The processing operations needed are compared with the capabilities of the factory by the evaluator. The evaluator may encounter different types of problems that relate to tolerances, geometry, available processes, and equipment. For representing capability and for reasoning with tolerance information, quality numbers are a useful tool.

Rapid prototyping and manufacturing techniques also fit into the product creator model. Rapid prototyping can function as a supplement to manufacturability evaluation, but it is not an alternative.

Chapter 4

Designing assemblies

In this chapter, details are provided on the actual design representation of parts and assemblies. Relations are introduced for modelling parts, assemblies, and tolerances. These relations are grouped into complex primitives to reduce complexity. The tool described eliminates some of the drawbacks of current modelling strategies.

4.1 Deficiencies of current product models

As explained in the previous chapters, current product creation processes and product models suffer from deficiencies that prohibit successful implementation of manufacturing evaluation. Summarising, they concern:

- Unsuitable primitives, such as volumetric entities and features are used.
- A focus on the final geometrical representation of an artifact instead of on the design process and functional relations.
- No integral use of tolerance modelling, parts modelling, and assembly modelling; tolerances are drawing attributes, while assembly modelling is separated.
- A lack of formal descriptions of processes and representations, resulting into ad hoc solutions, influenced by the possibilities or restrictions of existing tools.
- No clear separation between mental processes and formal product models, resulting into confusion on descriptions of the design process.

Most of these deficiencies were mentioned already. Some of them require exemplification however, especially the use of design primitives and tolerances.

Primitives

To provide details on the actual product representation, one needs suitable primitives. Currently, products are represented as a series of geometrical operations, of which the final result is kept. To support functional aspects and manufacturing, this is extended to (domain dependent) feature based representations, in which parametric shapes are provided with details (including ambiguous or irrelevant details). These shapes are not always able to make a distinction between critical (functional) and non-critical dimensions. The fact that dimensioning and locating a shape is separated is the major drawback of the use of volumes. It limits the freedom of modelling and prohibits the expression of functional requirements. It especially causes problems when tolerances are added because they have to be separated as well [Krom93]. The use of volumes also renders problems concerning design operations such as fillets, blends, and extrusions. The main problem however, is the fact that an approach based on volumes as primitives cannot properly be extended to assembly modelling, as mating relations between volumes are not defined.

Tolerances

In current design support systems, tolerances are underrated; representation, synthesis, as well as analysis. Tolerance representation makes sure that tolerances are an integral part of the product model, describing part of the product functions. Tolerance synthesis determines individual dimensions according to a sum dimension, while tolerance analysis investigates the effects of individual dimensions on a sum dimension. One distinguishes conventional tolerances and geometrical tolerances. Conventional tolerances specify the limits of dimensions. Geometric tolerances control size, form, orientation, and position. For geometrical tolerances international standards exist, like ANSI Y14.5M and ISO 1101. They are derived from the use of drawings. As a result, they appear in three dimensional models mainly as attributes. Besides that, basic principles and interpretations of tolerances vary, depending on the standard used [Henz95]. Geometrical tolerances partly meet the demand for better representations, but are still unsuited for reasoning with. In analysis, the tolerances are considered properties of a physical face. Consequently, tolerance relations are subject to a hierarchy, as depicted in Figure 4.1 (form $<$ orientation $<$ size).

Here, tolerance relations are considered constraints that should be met, which results into three new rules for analysis:

- Tolerance relations are not subject to hierarchy. A realised face conforms to hierarchy, but there is no functional of physical reason for constraints to do so.

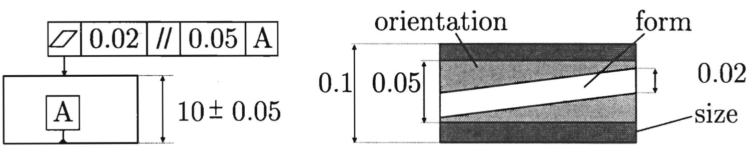


Figure 4.1: Hierarchy in tolerances according to ISO

- The strictest tolerance specified on a face is decisive. This enables reasoning with tolerances, but requires a mechanism to describe their mutual weight.
- No direction is indicated within a constraint. As opposed to what is customary, the designer does not indicate a datum plane in a tolerance relation.

In tolerance synthesis, most authors use cost reduction as a criterion, usually with fixed manufacturing processes and assuming a known output tolerance value [Liu91, Caga92, Gada94, Kris94, Salo95]. The cost functions have a hyperbolic shape and are usually combined with statistical methods (Monte Carlo based simulations), simulated annealing, or Taguchi loss functions. Functionality is a less prominent criterion, although it determines important dimensions. If this is observed, manufacturing costs are lower, as dimensions that are of less importance are not controlled (redundant and unnecessary tolerances, about 80% of all tolerances, are omitted [Ullm92]).

In tolerance synthesis, a designer needs to know how the tolerance relations relate to the functions, what tolerance value is needed to ensure functionality, and how individual tolerances can be adapted to the manufacturing capabilities. When establishing the correct tolerance relations, functionality is the major concern. For assigning individual tolerances, processes and costs are used, as depicted in Figure 4.2 [Holl95]. In both cases, quantitative statements are needed.

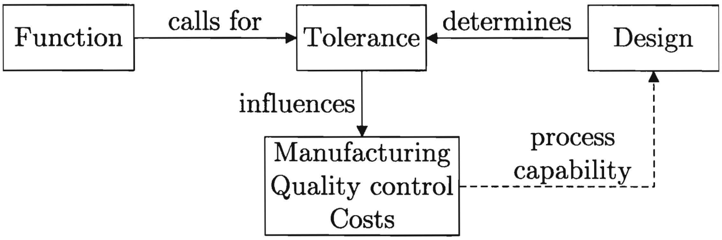


Figure 4.2: Propagation of tolerances

Functional requirements are linked to the topology of the mechanism in which a part operates and translate in terms of clearances and fits. Usually, the quantitative aspects of tolerances are based upon a designers' experience (often resulting into tolerances that are too strict), or worse, depend upon trial and error. While research for the relation in opposite direction is quite common (like the effects of machine guideway errors on machined surfaces), a limited number of quantitative studies on functional tolerances is found. Examples are press or shrink fits between a hub and a shaft, the Taguchi loss function, and models of the effect of geometrical errors on product's functionality [Kimu92, Taka93]. In the latter models, a guide mechanism is analysed. Information was obtained about the contacts between the two parts that can be used for creating tolerance information, which renders a more rational way of finding tolerances for assemblies of this type. The Taguchi loss function approach shifts the problem to finding correct loss functions and it also influences products that are within specification limits. This limits its use in finding quantitative tolerance information.

Tolerance representation and reasoning should enable tolerance synthesis based on functionality and manufacturing costs. If no quantitative relations are provided, designers should at least be given the opportunity to unambiguously and effectively describe the functional tolerances and to see their manufacturing consequences. How such a tolerance structure is built will be explained in the following sections.

4.2 The primitives of design

Relations are the smallest functional elements in design. They create the geometry of parts, model mating surfaces for assembly, and provide tolerance information. Design transformations serve as repositories for relations; a design transformation is applied by specifying a collection of relations. A designer does not draw the desired result of a step, but describes the relations, of which the geometry and the location of the parts is merely a result. This is demonstrated below. Figure 4.3a shows the volumetric approach that assumes a pre-defined shape (length, width, and height) to be located relative to the product co-ordinate system. Figure 4.3b depicts an approach in which the dimensions and the location are inferred from relations.

Relations connect variables, so specifying relations requires elements to serve as handles: the design primitives. Appendix C provides a review on entities that serve as examples when finding primitives and creating relations. The primitives used are planes (faces), cylinders (cylindrical faces), lines (edges), or points. In some cases, a sphere, a conical face, or a torus is used. Planes and lines are infinite or have boundaries. However, there is a crucial difference between faces and lines or points:

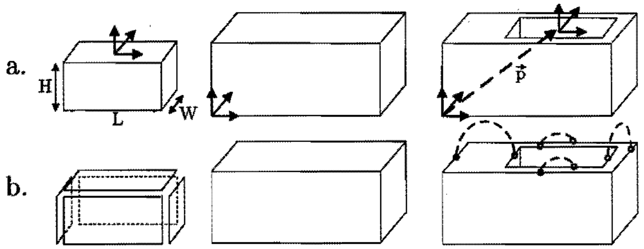


Figure 4.3: Dimensioning and locating geometrical entities

faces are physically present. Assembly operations involve mating faces, manufacturing processes generate faces, and faces are entities that can be directly measured by a co-ordinate measuring machine. Lines and points result from intersections, they cannot be manufactured or directly measured. Besides that, tolerances are mostly specified as face attributes (the relation between edge tolerances and functionality is questionable). Thus, faces are the design primitives.

Reference elements

As designers specify primitive elements by referring to other elements, they are called reference elements [Net94]. The prime set of reference elements consists of cylindrical faces and planar faces, as depicted in Figure 4.4. The elements are described using a vertex (point), a vector, and a radius (which may be infinite). Consequently, reference elements have infinite dimensions in one or two directions. Reference element intersections describe geometrical aspects of a design transformation.

The set of reference elements is extended with other types of faces to increase its applicability. Elements that could be added are a sphere, a cone, a torus, or free-formed faces. Adding elements requires a deliberation on complexity, functionality, and manufacturability. As free-formed faces are often influenced by aesthetics instead of functionality, tolerances cannot be defined, and the specification of relations poses problems, they are not applied here. Elements that add some functionality and for which tolerances are defined are a sphere and a cone. In some cases a line is also needed. The distance between two non-parallel faces cannot be determined, so when using a geometrical relation that specifies the angle between two faces the intersection has to be specified. Besides that, the centre line of a hole is often used, although it can be perceived as an abstraction of a cylindrical face. The additional reference elements are also described using a vertex, a vector, and a radius (cones need an attribute to describe the inclination), as depicted in Figure 4.4.

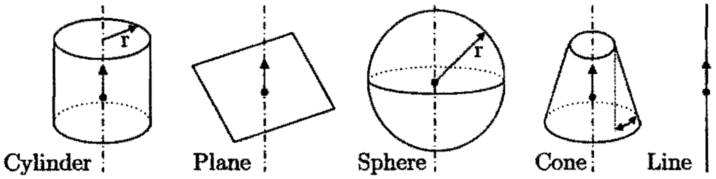


Figure 4.4: Reference elements

Geometrical relations

Reference elements are located by geometrical relations to determine the geometry of parts. Figure 4.5 shows an example of a design transformation that changes part geometry. It is assumed that the wall thickness of the resulting part is of functional importance. Relations position the planar faces forming the cavity walls parallel to the walls of a block at a relevant offset.

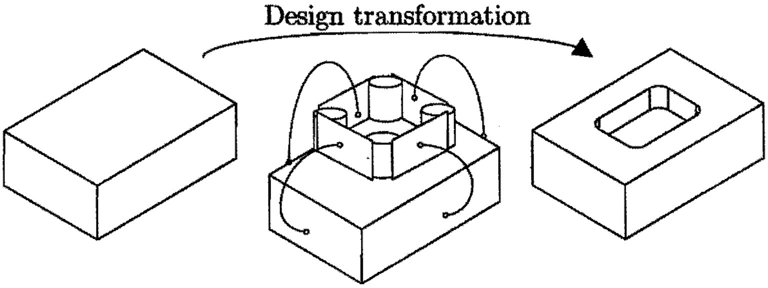


Figure 4.5: Design transformation example

Cylindrical faces, positioned relative to the planes, describe the cavity corners. A plane describes the cavity bottom. Intersecting all reference elements creates a volume, which itself is not a primitive. In the design transformation it is indicated that this volume should be subtracted, creating the result depicted. Different sets of relations render equal geometry. A designer decides what set of relations represents the functional requirements. Figure 4.6 depicts an alternative solution. The relations describing it are kept to make sure functionality is reflected in the product model.

A number of relations is used to create geometry: parallel faces, perpendicular faces, a face through two lines, two faces at an angle, or a line parallel to a line or face [Net94]. In a number of cases, multiple relations are needed to locate a single reference element, such as a line parallel to two faces. Sufficient relations should be

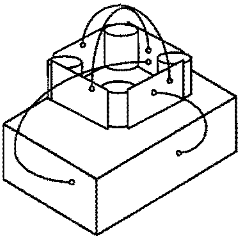


Figure 4.6: Alternative set of relations

provided to describe a volume. The implementation of the design support tool checks whether a correct design transformation is described.

Assembly relations

Assembly relations are not different from geometrical relations. The fact that the reference elements in the relations are from different parts is what makes them assembly relations. The reference elements in an assembly relation are available because the parts in a product were created using relations. Assembly relations specify mating relationships, such as against, align, or fits. Appendix C provides an overview of assembly relations found in literature. Here, some limitations are put on assembly relations. Only faces can be part of an assembly relation, lines or edges are not used. Assembly relations requiring elements that do not comply with the existing reference elements, such as point contacts, are explicitly rejected. This makes sure the relations have a basis in the physical world. Besides that, the relations used are fundamentally different, that is, relations that can be described as versions of other relations are omitted (fit, tight fit, and screw fit are not different relation types). Three basic assembly relations are chosen:

- alignment of two reference elements (faces or centre-lines)
- a reference element being against another (planar faces)
- a reference element fitting into another (cylindrical faces, cones, spheres)

An example of an assembly created using these relations is depicted in Figure 4.7. The number of relations between reference elements provided is sufficient to infer the locations of the parts. The remaining degrees of freedom are either functional (like in guideways or mechanisms) or do not pose any problems due to symmetry and sufficient stability (as in the example below, in which the peg may rotate).

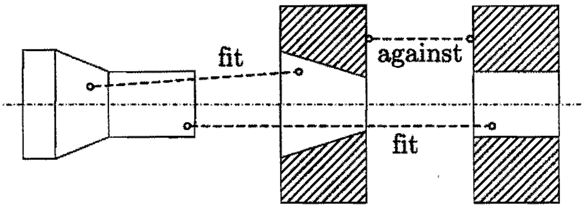


Figure 4.7: Assembly relations

Tolerance relations

Tolerance relations accompany geometrical relations or assembly relations to specify the allowable deviation of the nominal values. For example, when two planar faces are parallel, a parallelism tolerance may be added. It is also possible to create tolerance relations that do not accompany another relation. Adding tolerances to dimensions that were not obtained through relations however, will probably not have functional foundations. Besides to related tolerances, a face may have to conform to size restrictions, such as flatness or cylindricity. Both tolerance types are represented, with reference elements as handles.

The independence principle, the basic principle in the ISO standard, treats form and size tolerances independently. This makes them easier to meet and to inspect, but it is hard for a designer to get an insight in the overall implications. The envelope principle, used in the ANSI standard, states the limits of variations of form as well as size are in an envelope of certain width, located near the nominal face. The envelope principle is most used in industry and easier to interpret (especially in assembly operations) so it is used here. It is applied to all face reference elements, as depicted in Figure 4.8 for a cylindrical face and a planar face. To describe the envelope, either the median face and the tolerance zone width, or the tolerance zone limits are needed. The envelope is part of a reference element, which makes it applicable in relations and consistent with three-dimensional models. Tolerances that are surface properties rather than relations (roughness), are also part of the reference element.

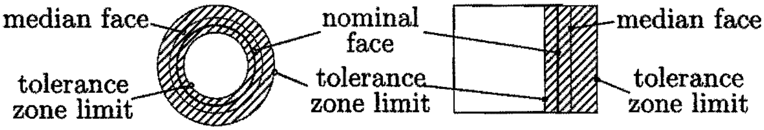


Figure 4.8: Envelope principle

The combination of all tolerances on a reference element describes the envelope, so their mutual weight should be defined. This is performed using the quality number, which was defined for representing capability. Extending this definition to other applications than fits is performed by adapting the characteristic length (L) value for each tolerance type [Vrie92]. For a parallellity tolerance for example, it depends on the lengths of the sides (A, B) and the distance between the faces (S).

$$i(L) = 10^{-3}(0.45L^{1/3} + 10^{-3}L) \text{ in which } L = \frac{1}{2}\sqrt{A^2 + B^2 + \left(\frac{1}{2}S\right)^2}$$

The equations are based upon empirical data. The quality number depends on face dimensions and the tolerance type. If multiple tolerances refer to one face (both related and unrelated tolerances), the quality numbers calculated for each tolerance are compared. The strictest one is decisive and thus describes the envelope of the reference element. The approach chosen is therefore different from an approach in which offsets on surfaces are combined to obtain a uniform tolerance zone for a volume [Requ83, Requ84], which makes the tolerances on the various faces dependent.

4.3 Product modelling

Modelling products with the concepts presented requires extra steps. Relations are not convenient to work with. Besides that, relations have to be reasoned with to derive a description of the artifact that is suited for further processing.

Abstractions

Designers tend to use abstractions, that is, pre-defined shapes (geometry) or the results of operations (assembly). In Figure 4.5, a designer perceives the collection of relations as a single entity (a pocket). The relations in Figure 4.7 form a peg-in-hole operation. Using abstractions reduces the complexity of the design process. Besides that, unstructured collections of relations are difficult to interpret during process planning, while for abstractions a set of pre-defined mappings can be found.

Therefore, a new set of entities is composed of reference elements. They are created to meet the need for abstractions, representing portions of geometry, operations, functionality, or combinations of these three aspects. This also creates the opportunity to specify complex primitives regarding assembly. Still, faces remain the primitives of design. Theoretically, an infinite number of primitives can be defined using reference elements. The number of primitives will be limited however, as any primitive should be provided with information about its manufacturing.

A design transformation can be represented as shown in Figure 4.9a. It is an operation requiring operators and operands, as depicted in Figure 4.9b. Obviously, the design state is an operand in design, while the result is the new design state. The design state may be the only operand, for instance when one of its properties is changed. Another possibility is the use of a design state and some assisting entity: the complex primitive. In a design transformation, the designer specifies the relations that result from applying such a primitive in a certain way. This interpretation of a design transformation is depicted in Figure 4.10.

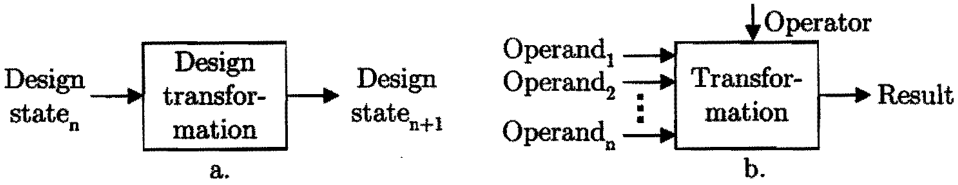


Figure 4.9: Transformations

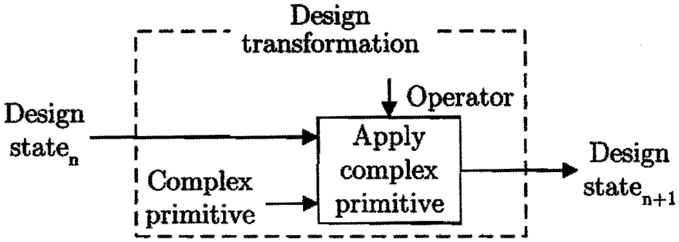


Figure 4.10: Design transformation and a complex primitive

Design transformations are unary or binary operations. In unary operations (properties changing, like annealing), only an operator is used. Unary operations still posses a manufacturing counterpart. Operations that require relations to be specified use a complex primitive and are therefore considered binary operations. Assembly operations use two design substates, connected using a complex primitive.

Primitives are distinguished for geometrical relations and assembly relations. Some relevant primitives, as well as the specification and verification of the relations, are discussed below. The permissible complexity of these primitives is limited. The prototype of the design support system implementing the concepts described here (refer to Chapter 6) has shown that designers should not be expected to specify more than five to ten relations, depending on the primitive. Complex primitives are used with several operators (like add and remove).

Geometrical primitives

A cylinder primitive specifies cylindrical holes or bosses. It is represented by a top plane, a bottom plane, and a cylindrical face, as depicted in Figure 4.11a (the elements are drawn as if they were finite). Some attributes are added to the primitive, such as whether it is threaded or if a tooltip is present. As the diameter of the cylinder cannot be inferred from relations, it is also specified. Figure 4.11b depicts a hole in a block. Although more familiar to a designer, it is just an instance of the cylinder primitive. To model it, the top plane and bottom plane coincide with two faces of the block, while the centre-line is parallel to two faces of the block.

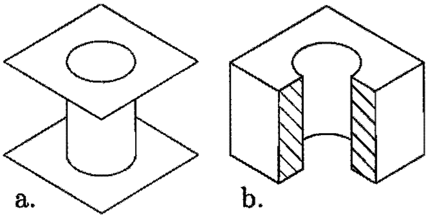


Figure 4.11: Cylinder primitive

Another example of a complex geometrical primitive is a slot. It can be described using two cylinders, a top plane, a bottom plane, and two side planes, as depicted in Figure 4.12a. Figure 4.12b depicts an instance of the slot primitive.

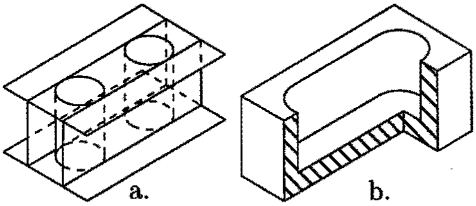


Figure 4.12: Slot primitive

In some cases, a location is difficult to specify, like in Figure 4.13a. A slot is positioned partly outside the material, which is difficult using the reference elements available. Creating a separate primitive is not desirable, so reference elements are added for locating purposes. They are not physically present, so they are called virtual reference elements. They can be of any of the reference element types. In a slot, three virtual planes are available, as depicted in Figure 4.13b. Virtual elements can not be used in tolerance relations or assembly relations.

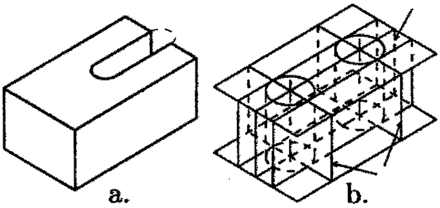


Figure 4.13: Virtual reference elements

Other complex primitives are presented in Figure 4.14; a rectangular pocket (a), a box (b) and a wedge (c). Curved wedges are represented by four planes and a cylinder (d). They are curved to the inside or outside, depending on the part of the cylinder used. Finally, a primitive is depicted for creating a blend (e).

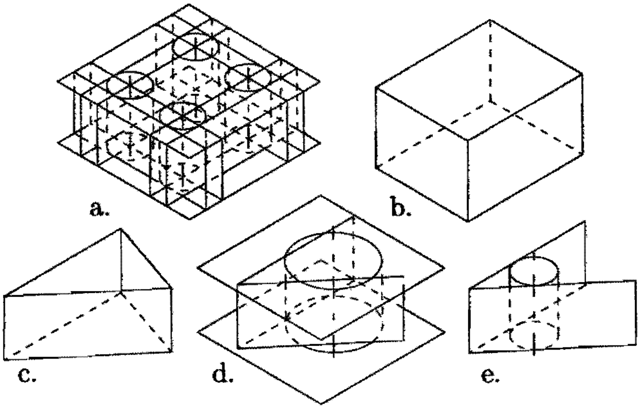


Figure 4.14: Other examples of primitives

The functional requirements determine the primitives needed. Although manufacturability is a key concept, it is not an argument for creating primitives. For instance, a slot and a rectangular pocket are separate primitives. This is not due to a difference in manufacturing, but to the differences in application. If a designer starts to use primitives to obtain results other than the originally intended ones, this indicates the need for another primitive. Improper use of primitives limits the use of functional relations and makes the mapping to manufacturing operations more difficult.

Assembly primitives

Complex assembly primitives completely describe a mate between two parts. As a result, fairly complex primitives are created, such as a mounting assembly primitive. In a primitive, attributes are added describing for instance lubrication conditions or glue. To comply with the permissible complexity of a primitive, the assembly should be static or comprise simple movements (sliding or rotating). Complex movements such as planetary gears are not used. Two examples of assembly primitives are depicted in Figure 4.15. A peg-in-hole primitive consist of a cylindrical boss fitting a cylindrical hole, while an against relationship is used to specify the position in vertical direction. When connecting two parts with a bolt, the holes are aligned, the faces of the two parts are against each other, the bolt fits into the holes, and so on.

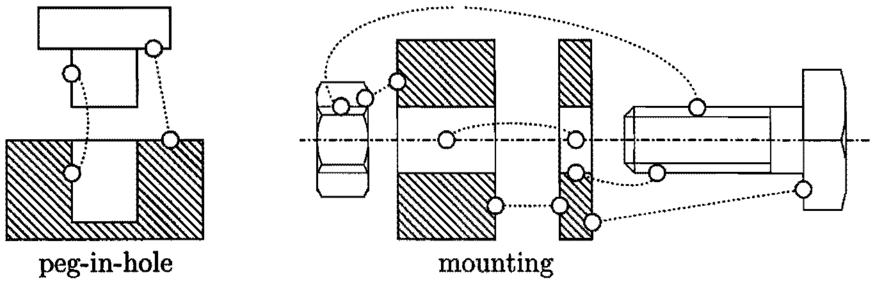


Figure 4.15: Assembly primitives

A set of two relations is the smallest complex primitive that can be created. Extended complex primitives for instance concern combinations of multiple peg-in-hole or mounting primitives. Mapping them onto assembly operations usually is not very different from the mapping of the simple ones. The manufacturability checking of a extended primitives is more difficult however, as the separate mating conditions tend to interact (like the double peg-in-hole problem).

Modelling example

Figure 4.16 depicts an example assembly consisting of a cylinder, a piston, and a piston rod [Net96]. Various geometrical primitives are used, such as cylinders, rectangular pockets and slots. The parts are connected using fits of cylindrical reference elements and alignments or against relations (a peg-in-hole primitive). Part of this design is examined to demonstrate the use of relations in a design transformation. A slot in the side of the piston is specified using relations. To ensure proper movement of the piston, tolerance relations are added.

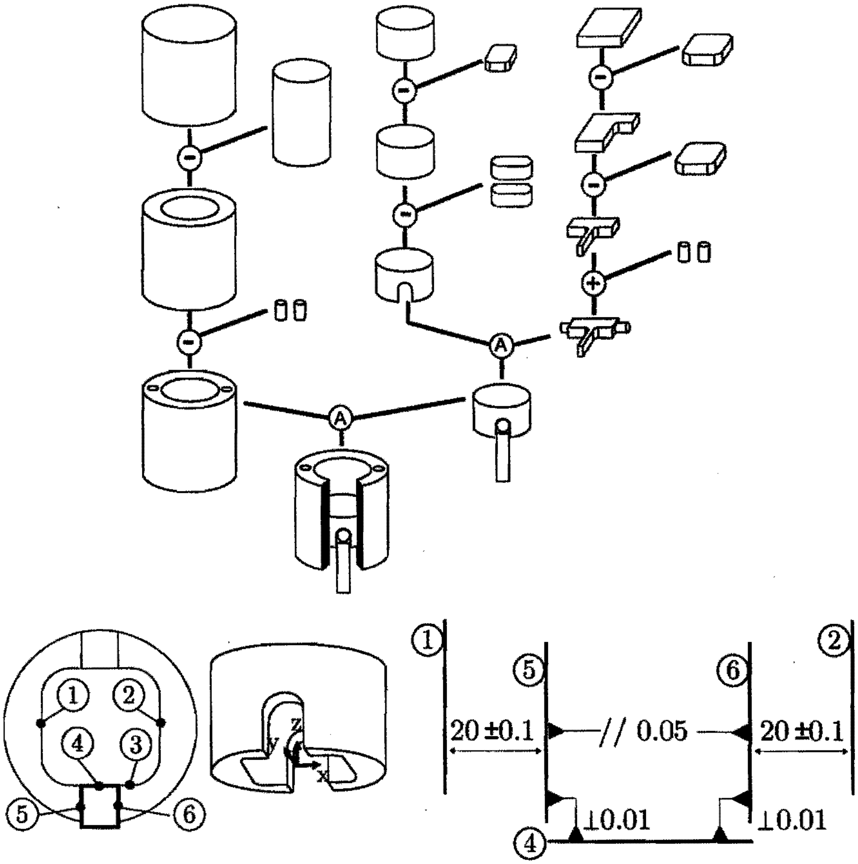


Figure 4.16: Typical example

Propagation

When relations are specified, the resulting geometry is calculated, the positions of parts are found, and tolerances are reasoned with (tolerance analysis and synthesis). Generally, a set of variables (reference element locations) and a set of constraints (reference element relations) is given. Finding a set of assignments for each variable consistent with the constraints is called constraint satisfaction.

A prominent technique for determining the location of a shape in geometrical design is analysis of the degrees of freedom. However, when using geometrical relations to specify the location as well as the dimensions of a volume, this is insufficient. Besides that, a distinction is often made between primary and secondary elements in a constraint, that is, the secondary element moves to meet the constraint. As there is no physical or functional motivation for such a distinction, the reference elements in a relation are equally important here.

Propagation of geometrical relations

Geometrical relations specify a relatively complex network of constraints. Such constraint networks are often represented as graphs using the variables as nodes and the constraints as edges, or with nodes being a variable or an operator. The latter is shown in Figure 4.17a, a constraint network representing A*X+A*Y=Z [From92]. To find mechanisms for solving constraints, one may refer to so-called constraint satisfaction planning (CSP) techniques, such as propagation and term rewriting. Propagation involves activating any operator in the network with enough information about its entries. This is repeated until no more nodes can be activated. Term rewriting rewrites the complex network into simpler sub-graphs. Figure 4.17b depicts the result of this technique using the example.

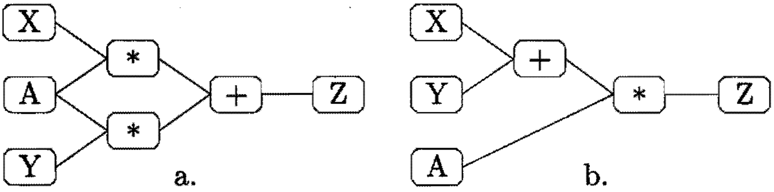


Figure 4.17: Constraint networks

Propagation is widely used, easy to implement, and not limited to numerical problems. However, no mechanism is provided for nodes having multiple outgoing links, that is, A cannot be inferred if X, Y, and Z are known. In geometrical relations, multiple links from a node are quite common, so the more powerful graph transformation or term rewriting technique should be considered.

A CSP approach used by Arbab and Wang to reason about geometrical constraints uses sets of constraints that form a pattern in a constraint network [Arba90]. Such sets, so-called clichés, are recognised and correspond with a special meta-operation. A set of operations and clichés is provided to create practical applications.

A combination of the above techniques renders a solution to the propagation problem encountered here. As all copies of a primitive share the same set of geometric constraints, these sets are analysed to find specific constraint resolving algorithms. Geometry is specified in three orthogonal directions, so the reference elements in a complex primitive are grouped into three orthogonal directions. For each of these directions, it can be easily determined what combinations of reference elements are sufficient to obtain all information needed. An example of this decomposition is depicted in Figure 4.18, a box decomposed into three sets of planes.

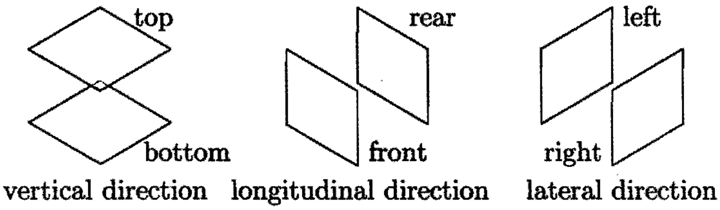


Figure 4.18: Decomposition

Whereas other constraint satisfaction techniques aim at being independent of the nature of the constraints, this method explicitly uses knowledge about the constraints. A drawback that results from this concerns the fact that the solving algorithm should be adapted to each primitive, instead of being generally applicable. The decomposition principle upholds for any primitive, although the amount of information that is needed may vary (a cylinder primitive for instance is less complex due to symmetry).

Figure 4.19 explains the constraint satisfaction technique. The information needed consists of the orientation of the planes and a dimension in each direction, that is, the distance between the planes. The reference elements available are the six planes of the box and all edges. Some combinations of reference elements are depicted on the left (not all combinations possible). If an element is drawn as a continuous line or shaded plane it is specified by the designer. For each combination it is indicated whether the information content is sufficient or not. Furthermore, two decomposition examples are depicted. It should be noted that some elements (edges) are used in multiple directions, that is, specifying one reference element may provide information for two of the orthogonal directions in a primitive.

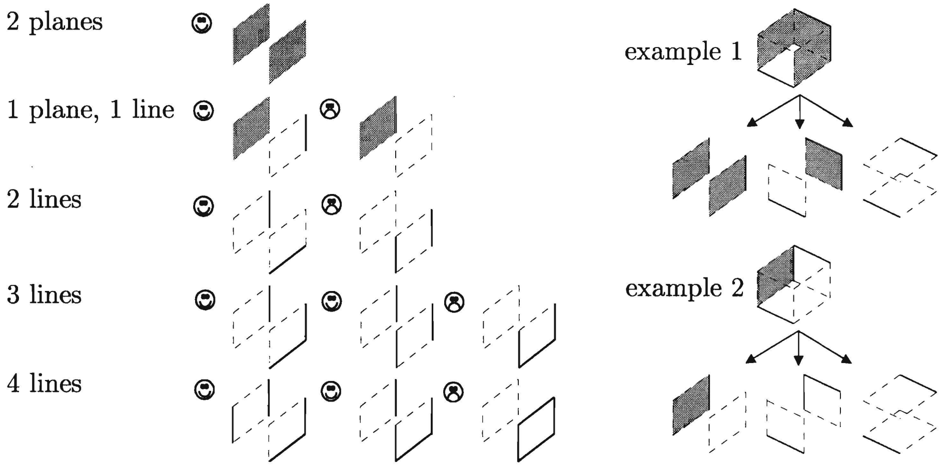


Figure 4.19: Decomposition of a box primitive

After a designer decides what geometrical primitive is to be used in a design transformation, the available reference elements are presented. The relations that describe the primitive are created and propagated according to a three step procedure:

1. A reference element is specified using one of the available methods (depending on the type of element), such as a plane that is parallel to another plane. As a result, a reference element of the primitive has a fixed location and orientation.

2. The properties of the reference element are checked against the internal constraints. For instance, the opposite sides of a box should be parallel.

3. The element that was specified is added to the existing elements of the primitive. It is checked whether sufficient information is available to infer the properties of a shape. If the information is available, the shape is used to create a design transformation. If not, more reference element relations are added.

When using this procedure, no over-constrained situations can occur. The reference elements are specified one by one. When a relation is added that completes the information needed, no further relations are accepted. Once a correct number of reference element relations has been established, the resulting shape can be inferred using vector calculus. Examples are the calculation of length, width, and height by determining the distance between the point describing opposite reference elements or determining the orientation of a shape from the reference element vectors. The inferred properties (volume and location) are used for visualisation and process planning. More details are provided in Appendix C.

Propagation of assembly relations

Assembly relations are not used for dimensioning purposes. Therefore, analysis of the degrees of freedom is a suitable technique. Each of the assembly relations leaves some degrees of freedom, so the degrees of freedom that result from applying a primitive can be deduced from the relations it contains. As the primitives completely specify a mate between two parts, the remaining degrees of freedom depend upon the primitive involved. Figure 4.20 depicts the degrees of freedom of assembly relations. Figure 4.20a shows the degrees of freedom that exist. Figure 4.20b shows what is left after specifying an against relation, Figure 4.20c depicts the result of a fit relation, and Figure 4.20d and e depict alignment of a face and a line respectively. The degrees of freedom are relative, independent of assembly direction, as opposed to for instance [Liu91], where it is claimed that the degrees of freedom depend on which object is relative with respect to which other object.

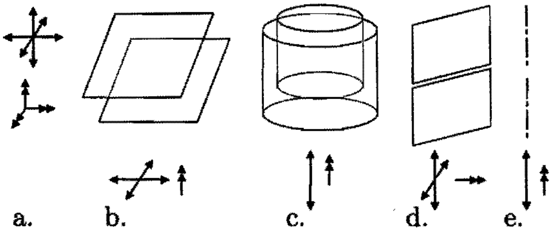


Figure 4.20: Degrees of freedom in assembly relations

From the assembly relations, the degrees of freedom of a complex assembly primitive are deduced. A peg-in-hole primitive for instance, is a combination of Figure 4.20b and c, leaving a rotation about the peg axis. In case of a square peg, the against relation is used five times, thus some degrees of freedom are determined multiple times. Such over-constrained situations indicate possible assembly problems. This results into some extra considerations, such as analysis of clearance during process planning. If needed, the geometry is modified to add some chamfers.

Relative locations of parts are calculated using transformation matrices [Lee85b, Ko87, Roch87]. Generally, the part that is added to the assembly translates and rotates to meet the constraints. As a mate between two parts is completely specified using a complex primitive, careful selection of the relations in the primitive prevents over-constrained situations concerning the relative location. In case of situations such as mentioned above (peg-in-hole), the variation in location that occurs when multiple relations are propagated should be within the specified tolerance limits.

The primitive also determines the assembly directions. This is explained in Figure 4.21. Every relation comprises a set of possible assembly directions. This set can be expressed as a segment of a sphere [Woo90]. An against relation for instance, can be assembled from all directions in a set described by half a sphere. Combining the segments for all relations in a primitive renders the assembly directions.

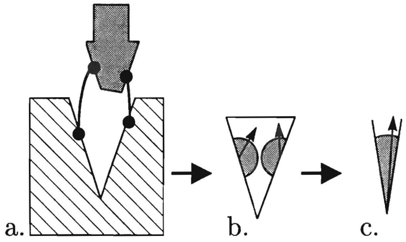
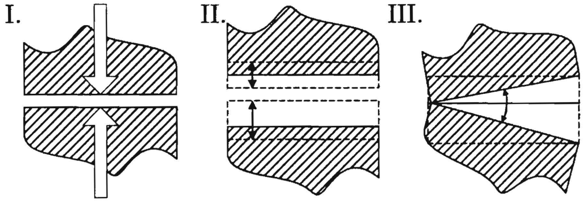


Figure 4.21: Assembly directions

Propagation of tolerance relations

Some aspects of tolerance propagation were assessed, such as their mutual weight and manufacturing consequences. In assembly, the consequences of part tolerances for the assembly are found, or part tolerances are derived from an assembly tolerance.

Complex assembly primitives are functional elements, so designers can indicate parameters and deviations for these parameters that express the acceptable behaviour of the assembly. Such parameters are the maximum deviation of position and orientation of a guideway, the maximum deviation of transmitted torque in a press fit, or the play of a fit for a rotational element. These parameters are part of a primitive. The contributions of all relations are analysed using a worst case scenario. The tolerance zones of the faces participating in the relations are used to find the maximum deviations possible. Figure 4.22 depicts the maximum deviation of the relative orientation of two faces in an against relation. The contributions of all relations in a primitive are used to derive the reverse effect of parameters upon the individual relations. This renders the tolerances for each relation in the primitive.



I. Relation: against
II. Tolerance zones
III. Worst case

Figure 4.22: Worst case analysis of the against relation

For distributing an assembly tolerance, that is, finding the part tolerances, it is generally stated that tolerance values are equally distributed among the parts. This results in equality of manufacturing processes, which is a good criterion to find the minimum manufacturing costs. This type of propagation renders three problems. First of all, a part may participate in multiple assembly relations. As a result, equal distribution of tolerances does not necessarily render equality of manufacturing processes within a part. Besides that, the manufacturing effort is also determined by the dimensions of a face, so equal distribution may still result into different processes. Thirdly, the faces are often created using different manufacturing processes (like a hole and a shaft), for which the limits of process capability are different.

The tolerances of the assembly relations are therefore distributed using quality numbers. This enables a comparison of the processes needed for alternative distributions of the tolerance value. It uses the actual factory capability and the actual part geometry. For finding the minimal costs, the quality numbers are kept equal for the faces in assembly relations. If these faces are also in tolerance relations of a part, equality is also observed within the part. If a face is in multiple tolerance relations, the strictest one is decisive. The face is made to satisfy this tolerance value, that is, too accurate when the other tolerance relations are concerned. The tolerance zones are then redistributed. The part of a tolerance zone that is left over is used to decrease the manufacturing effort of the other face, as will be explained in an example later [Net96, Vrie96].

The assumption behind this concerns the manufacturing costs. These costs depend upon manufacturing effort. Due to the non-linear relation between manufacturing effort and the quality number, the extra effort needed for raising a quality number is not compensated by lowering the quality number of the other face by the same amount. Thus, equal quality numbers will result into the lowest total effort and therefore the lowest total cost. This renders two additional benefits:

- An indication of the costs that are associated with a decision, so hints can be provided for lowering them. Besides that, obsolete tolerances are found. Some tolerances are automatically met if other, stricter, tolerances are met.

- After tolerance propagation, reference elements have median faces. The median faces describe a non-nominal geometry. This geometry is used for further processing, as occurs in real-world cases. One has to be careful however, not to create volumes in which the faces do not connect.

Propagation example

To demonstrate tolerance propagation, the example of Figure 4.16 is further examined. Figure 4.23 depicts relations that, together with a value for the depth and relations for the vertical location, specify a slot primitive. Table 4.1 presents the associated quality numbers. For each tolerance relation, the quality numbers of the faces are equal, while the sum of the tolerance widths equals the tolerance latitude (0.2, 0.05, and 0.01). As an example, Figure 4.24 depicts the quality number found for the 0.05 parallellity tolerance (8.7). Faces 5 and 6 measure 30 times 15 millimetres and are 20 millimetres apart, so characteristic length L=18.

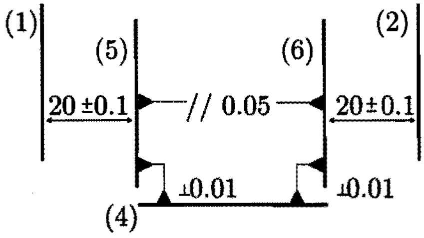


Figure 4.23: Example relations

	face 1	face 2	face 4	face 5	face 6
20 ± 0.1	9.6			9.6	
20 ± 0.1		9.6			9.6
//0.05				8.7	8.7
⊥ 0.01			5.3	5.3	
⊥ 0.01			5.3		5.3

Table 4.1: Quality numbers

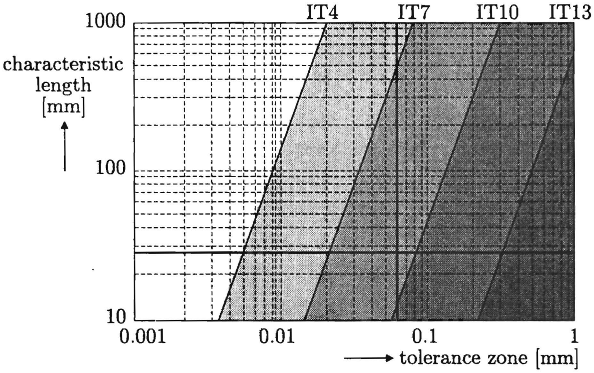


Figure 4.24: Determining a quality number

As the strictest tolerance is decisive, the perpendicularity tolerances are critical. The parallellity tolerance is obsolete. Faces 5 and 6 are manufactured to meet quality number 5.3, creating a tolerance zone that is smaller than needed for the size tolerances: 0.01 millimetres. The remaining part, 0.2-0.01=0.19 millimetres, is allocated to faces 1 and 2, as depicted in Figure 4.25. This is permitted, as tolerances are constraints that should be met, so the distribution of the value among the faces is not established. Here, faces 1 and 2 are manufactured to meet quality number 11 instead of 9.6, without failing the size tolerance.

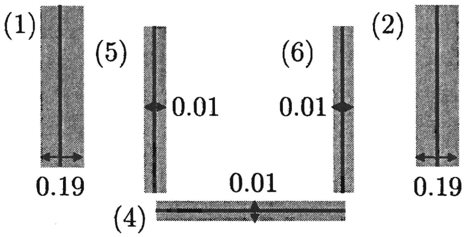


Figure 4.25: Example tolerance zones

Another detail is depicted in Figure 4.26: the assembly relations (fit relations and against relations) between the piston and the piston rod. Accurate machines with moderate speeds require a close running fit (ANSI standard fit RC4). For a diameter of 20 millimetres, a play of about 0.07 millimetres is distributed among the faces. As both faces are the same size, equal manufacturing effort results into a tolerance zone of 0,035 millimetres each (IT8). The assembly relations leave two degrees of freedom, a rotation and a movement in vertical direction (the assembly direction).

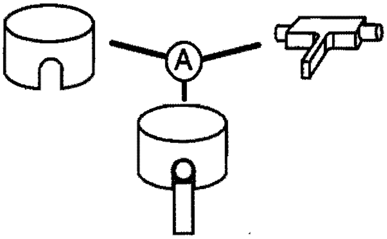


Figure 4.26: Example assembly relations

A designer might reconsider the perpendicularity relations. By using a tolerance value of 0.02 instead of 0.01, faces 4, 5, and 6 need no grinding operation, saving time and costs (but functionality is changed). It is also possible to remove these relations and specify a stricter parallellity relation (requiring quality 5.3, so functionality is not changed).

4.4 Design grammar

For improving the product creation process, parts and products have to be represented in a consistent and meaningful way. A set of entities was created to represent the geometrical design process and the product model. The term ontology of design has been taken up to designate the building blocks out of which models are made; the basic level of knowledge representation [Grub92]. Differences in abstraction and views prevent a shared ontology, so a domain is chosen. Informally, ontology then specifies the use of a domain-specific language; sets of formally described terms and meanings [Cutk93]. This is closer to a grammar or formal language [Brow95].

> *A grammar is a formal specification of a set, consisting of a set of primitives and a finite set of productions which specify transformations of those primitives. By recursive application of the rules, a grammar may be used to generate members of the set. By applying the rules in reverse, it may be used to recognise members of the set.*

A grammar is a 4-tuple $G =< N, T, S, P >$. A set of non-terminal symbols N, a set of terminal symbols T ($N \cap T = \varnothing$), a start symbol S, and a set of productions

P. Language G is the set of all strings derived from the start symbol, consisting of terminal symbols. The meaning of a structure in a language (semantics) is determined by the entities it refers to and the relationships it asserts between them. When using formal specification languages, the product model is (and remains) unambiguous, verification of correctness becomes easier, and the solution of implementation tasks becomes easier. The objective here is to prove that the concepts presented are consistent and unambiguous. Furthermore, a design language forms the basis of an implementation. It is a specification for application builders or serves as a language for representing models of artifacts.

Syntax

The notation of the grammar is in Extended Backus Naur Form (EBNF). A single structure name (non-terminal) is on left hand side and symbols or other structure names are on the right hand side. One must always be able to arrive at a string of terminals that cannot be broken down [Loud93]. The symbols used are in Table 4.2.

structure	$< a >$	group	$(a\ b)$
definition symbol	::=	repetition	$\{a\}$
selection	$a \mid b$	optional structure	$[a]$

Table 4.2: EBNF notation

The top level of the syntax definition concerns the product model, a collection of design states consisting of one or more sub-states connected by assembly transformations. A sub-state is formed using geometrical transformations and a material. Material properties are changed using material transformations. This renders:

$$\begin{array}{rcl}
\langle Product\ Model\rangle & ::= & \{\langle Design\ State\rangle\} \\
\langle Design\ State\rangle & ::= & \{\langle Design\ Substate\rangle\} \\
 & & [\{\langle Assembly\ Transformation\rangle\}] \\
\langle Design\ Sub - State\rangle & ::= & \{\langle Geometrical\ Transformation\rangle\} \\
 & & \langle Material\rangle\ [\{\langle Material\ Transformation\rangle\}] \\
\langle Assembly\ Transformation\rangle & ::= & \langle Assembly\ Primitive\rangle \\
\langle Geometrical\ Transformation\rangle & ::= & \langle Geometrical\ Primitive\rangle \\
 & & \langle Geometrical\ Operator\rangle \\
\langle Material\ Transformation\rangle & ::= & \langle Surface\ Treatment\rangle\ | \\
 & & \langle Material\ Treatment\rangle
\end{array}$$

At a lower level, the primitives and relations are defined as structures in the design language. The right hand symbols are not terminals yet. The remaining details of the syntax definition are found in Appendix D.

⟨*Assembly Primitive*⟩ ::= ⟨*Assembly Primitive Type*⟩ {⟨*Relation*⟩} [{⟨*Attribute*⟩}]
⟨*Geometrical Primitive*⟩ ::= ⟨*Geometrical Primitive Type*⟩ {⟨*Relation*⟩} [{⟨*Dimension*⟩}] [{⟨*Attribute*⟩}]
⟨*Assembly Primitive Type*⟩ ::= *Peg in hole* | *Mounting* | ...
⟨*Geometrical Primitive Type*⟩ ::= *Box* | *Cylinder* | *Slot* | ...
⟨*Relation*⟩ ::= ⟨*Reference Element*⟩ ⟨*Relation Operator*⟩ [⟨*Dimension*⟩] ⟨*Reference Element*⟩

The syntax of the language describes the symbols used. The language is the set of strings that can be derived, consisting of terminal symbols. Productions or grammar rules derive strings of the language. The semantics are determined by the entities the strings refer to. By demonstrating the derivation of language constructs that accompany the design of an example product, the semantics are clarified.

Semantics

A simple assembled product illustrates the use of the design grammar, as depicted in Figure 4.27. The product, a peg inserted into a hole in a block, is described using four steps. A geometrical description of the design states (DS) involved is depicted.

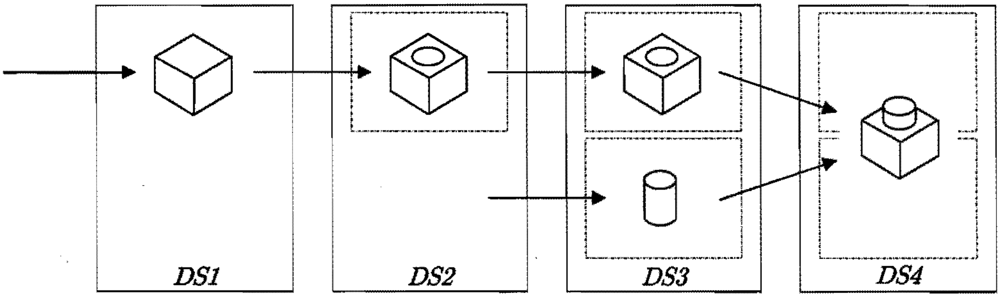


Figure 4.27: Example product design

The product model grows from one to four design states, as a new design state is added to the product model instead of replacing the previous one. Design states 3 and 4 consist of two design substates. Even though assembly relations are defined in the last design state, the two substates remain present. The productions describing the first step concern a product model containing a single design state (DS1). A block is created by adding a complex primitive of type box. The designer selects a *Box* to be applied and enters the dimensions to define the first geometrical transformation in the design substate. To completely define the substate, the material ($C35$) is entered. As the box is the first primitive present, the relations are only internal relations, describing the faces of the box (parallel and perpendicular faces). These relations are not elaborated.

$$\begin{array}{lll}
\langle \mathit{Product\ Model} \rangle & \rightarrow & \langle \mathit{Design\ State} \rangle \rightarrow \langle \mathit{Design\ Substate} \rangle \\
 & \rightarrow & \langle \mathit{Geometrical\ Transformation} \rangle \ \langle \mathit{Material} \rangle \\
 & \rightarrow & \langle \mathit{Geometrical\ Primitive} \rangle \ \langle \mathit{Geometrical\ Operator} \rangle \ C35 \\
 & \rightarrow & \langle \mathit{Geometrical\ Primitive\ Type} \rangle \ \{\langle \mathit{Relation} \rangle\} \ \mathit{Add}\ C35 \\
 & \rightarrow & \mathit{Box}\ \{\langle \mathit{Relation} \rangle\} \ \mathit{Add}\ C35
\end{array}$$

The second step is more interesting. At this point, the product model consists of two design states ($DS1$ and $DS2$). Each of these design states still consists of a single substate. The first design state does not change. The second design state describes the fact that a part of the product was provided with a hole, as depicted in Figure 4.28. Consequently, a geometrical transformation is added to the second design substate.

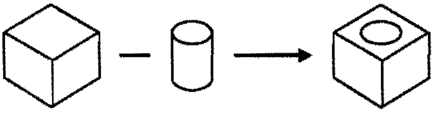


Figure 4.28: Detail of the second design state

$$\begin{array}{lll}
\langle \mathit{Design\ State} \rangle & \rightarrow & \langle \mathit{Design\ Substate} \rangle \\
 & \rightarrow & \langle \mathit{Geometrical\ Transformation} \rangle \\
 & & \langle \mathit{Geometrical\ Transformation} \rangle \\
 & & \langle \mathit{Material} \rangle \\
 & \rightarrow & \ldots
\end{array}$$

The designer needs a cylindrical hole. Therefore, the geometrical transformation consists of a primitive of type *Cylinder* and a *Remove* operator.

$$\begin{aligned}
\langle \mathit{Geom.\ Transformation}\rangle &\rightarrow \langle \mathit{Geometrical\ Primitive}\rangle\ \langle \mathit{Geometrical\ Operator}\rangle\\
&\rightarrow \langle \mathit{Geometrical\ Primitive\ Type}\rangle\ \{\langle \mathit{Relation}\rangle\}\\
&\qquad \mathit{Remove}\\
&\rightarrow \mathit{Cylinder}\ \{\langle \mathit{Relation}\rangle\}\ \mathit{Remove}\\
&\rightarrow \ldots
\end{aligned}$$

The primitive consists of a number of relations, one of which is internal: the top and the bottom face are parallel. In this case, it is also used to define the cylinder depth. The other relations are expanded by the production rules when the information from the designer is available. A *Relation* was described in the syntax as:

$$\begin{aligned}
\langle \mathit{Relation}\rangle &\rightarrow \langle \mathit{Reference\ Element}\rangle\ \langle \mathit{Relation\ Operator}\rangle\\
&\qquad \langle \mathit{Dimension}\rangle\ \langle \mathit{Reference\ Element}\rangle\\
&\rightarrow \langle \mathit{Reference\ Element\ Type}\rangle\ \langle \mathit{Virtual}\rangle\ \langle \mathit{Vector}\rangle\ \langle \mathit{Vertex}\rangle\\
&\qquad \langle \mathit{Envelope}\rangle\ \langle \mathit{Relation\ Operator}\rangle\ \langle \mathit{Dimension}\rangle\\
&\qquad \langle \mathit{Reference\ Element\ Type}\rangle\ \langle \mathit{Virtual}\rangle\ \langle \mathit{Vector}\rangle\ \langle \mathit{Vertex}\rangle\\
&\qquad \langle \mathit{Envelope}\rangle
\end{aligned}$$

As determined in the primitive definition, only a limited number of combinations of relations can occur. The number of relations needed depends on the relations. The relations used to locate and dimension the cylinder are presented in Table 4.3.

Box RE	Cylinder RE	Relation operator	Dimension
top (face)	top (face)	//	(coinciding)
front (face)	cylinder (face)	//	50
right (face)	cylinder (face)	//	50
front (face)	cylinder (face)	// tolerance	0.1
right (face)	cylinder (face)	// tolerance	0.1

Table 4.3: Relations

The second relation places the centre line parallel to the front face of the box. At this point, some productions are identified. The first line represents the first reference element, a *Plane*, not virtual (*False*), defined by a vector $(1,0,0)$ and a vertex $(100,0,0)$. The second line represents the relation operator parallel (//) and the dimension: 50 millimetres. The other reference element is represented in the last line, a *Cylinder*, not virtual, of which the details are to be determined.

$$\begin{aligned}\langle Relation\rangle \rightarrow\ & (Plane\ False\ (1,0,0)\ (100,0,0)\ \langle Envelope\rangle)\\ & //\ 50\\ & (Cylinder\ False\ \langle Vector\rangle\ \langle Vertex\rangle\ \langle Envelope\rangle)\end{aligned}$$

The fact that a cylinder is parallel to the face implicates that its vector and vertex are $(0,1,0)$ and $(50,0,0)$ respectively. The envelope is derived from the tolerance relations. The details that were not determined above are now known, such as the envelope:

$$\begin{aligned}\langle Envelope\rangle &\rightarrow (\langle Median\ Face\rangle\ \langle Dimension\rangle)\mid(\langle Boundary\rangle\ \langle Boundary\rangle)\\ &\rightarrow \langle Vector\rangle\ \langle Vertex\rangle\ \langle Dimension\rangle\\ &\rightarrow \langle Vector\rangle\ \langle Vertex\rangle\ \langle Dimension\rangle\\ &\rightarrow (0,1,0)\ (50,0,0)\ 0.1\end{aligned}$$

To illustrate assembly modelling, the fourth step is described. The product model consists of four design states at this point. The fourth design state contains two design sub-states, connected with a complex assembly primitive of type peg in hole; a fit relationship and an against relationship (operators represented by $\circledcirc$ and $\veebar$).

$$\begin{aligned}\langle Design\ State\rangle &\rightarrow \langle Design\ Substate\rangle\ \langle Design\ Substate\rangle\\ &\ \langle Assembly\ Transformation\rangle\\ &\rightarrow \ldots\\ \langle Assembly\ Transformation\rangle &\rightarrow \langle Assembly\ Primitive\rangle\\ &\rightarrow \langle Assembly\ Primitive\ Type\rangle\ \{\langle Relation\rangle\}\\ &\rightarrow Peg\ in\ hole\ \langle Relation\rangle\ \langle Relation\rangle\\ &\rightarrow Peg\ in\ hole\\ &\ \langle Reference\ Element\rangle\ \langle Relation\ Operator\rangle\\ &\ \langle Reference\ Element\rangle\ \langle Reference\ Element\rangle\\ &\ \langle Relation\ Operator\rangle\ \langle Reference\ Element\rangle\\ &\rightarrow Peg\ in\ hole\\ &\ \langle Reference\ Element\rangle\ \circledcirc\ \langle Reference\ Element\rangle\\ &\ \langle Reference\ Element\rangle\ \veebar\ \langle Reference\ Element\rangle\\ &\rightarrow \ldots\end{aligned}$$

Reflection about the results

The design grammar shows a consistent and coherent set of symbols. The entities are suited for describing mechanical designs. Relations are used as a basic mechanism to create a product model, which enables the expression of functionality. Relations and reference elements are the primitives of design, other symbols can always be expanded to relations and reference elements using productions. At the level below the reference elements, only mathematical entities like dimensions and vectors exist.

The design grammar and the entities the strings refer to resolve the need for a design process oriented description of a product. Parts modelling and assembly modelling are integrated and tolerances are an inseparable part of the description. The rules defining the language are presented to the designer as design options. A process planner, inspection planner, assembly sequence planner, and other applications could be implemented as modules. These modules interpret the product model that was described using the design language. The generation of the geometric model can also be performed by such a module. This proves that geometry is derived from the product model instead of being a key element.

The grammar is only applicable in creating a mechanical design; none of the symbols provides any information on the manufacturing aspects. For mapping design operations onto manufacturing operations (the planners mentioned above), a similar set of symbols is needed for representing manufacturing entities.

Additional symbols can be added to the design grammar to increase the practicality of the language. In grammars, this is called syntactic sugar [Loud93]. The number of additional symbols should be limited however to avoid abrogating the benefits of using a language. A galore of additional symbols and entities endangers the consistency and clarity of the grammar. Suggestions of symbols that may be added are the names of reference elements or entities for storing the geometric properties that were inferred from the relations.

4.5 Summary

In this chapter, relations were presented as the smallest functional elements in design. A design transformation is applied by specifying a collection of relations. A relation connects two or more variables, so specifying a relation requires elements to serve as handles. These handles, called reference elements, are the design primitives: cylindrical faces, planar faces, spheres, cones, and lines.

Geometrical relations determine the geometry of parts by locating reference elements. Relations like parallel faces, perpendicular faces, a face through two lines, or two faces at an angle are used. Assembly relations, such as the alignment of two reference elements, a reference element being against another, and a reference element fitting into another, describe a mate between parts. Reference elements were also applied to model tolerance relations. For this, the envelope principle was used.

Because relations are not really convenient to work with, pre-defined shapes (for geometry) or the results of operations (in assembly) were introduced as abstractions. These abstraction, complex primitives, do not suffer from the drawbacks of features because they are based upon different principles. Primitives like a cylinder, a slot, a rectangular pocket, a blend, a peg-in-hole connection, and a mounting primitive were provided. An example was given to demonstrate their application.

After specification, the relations are propagated through the product model. For geometrical relations, a special constraint satisfaction planning technique was provided, while analysis of the degrees of freedom is a suitable technique for assembly relations. Propagation of tolerances is composed of tolerance synthesis and tolerance analysis. Tolerance analysis uses the quality numbers. For tolerance synthesis, cost reduction is used as a criterion for distribution, when functionality has determined the tolerance types and values.

To prove that the concepts presented (design transformation, design state, relation, complex primitive) are consistent and unambiguous, they were described as elements of a formal specification language, which is also used for implementation purposes.

The next chapter deals with the manufacturing operations that are used to create the products specified using the tools presented in this chapter. As the manufacturing of parts is sufficiently covered by the authors mentioned in the previous chapters, the focus will be on assembly operations.

Chapter 5

Manufacturing assemblies

Design operations are mapped upon manufacturing operations to enable their verification and to create a product. In connection with the previous chapter, this leads to a design grammar being read and translated into a manufacturing grammar. Such a mapping is possible, provided that the entities that are referred to and the processes that are described are known. A lot of effort has been put in describing the characteristics of parts manufacturing, but the entities and processes used in assembly are less clear. The first part of this chapter concentrates on finding the characteristics of assembly process planning. It is concerned with finding suitable assembly operations, validity checking, and the assembly sequence. When these aspects are covered, the actual mapping of design operations onto manufacturing operations is possible.

5.1 Assembly operations

Assembly process planning deals with the generation of manufacturing information needed to transform a group of loose parts into an assembled product. Four levels of abstraction in assembly process planning are identified [Heem90]:

- The batch level considers a batch of products, for which assembly equipment is made available.
- The product level considers one product, assembly is a series of join operations.
- The part level considers one part, assembly is a sequence of actions bringing it from the loose to the joined state.
- The primitive level considers a task, executed with actuators and sensors.

The product level, part level, and part of the primitive level are considered. No clear boundary is found here between the part and primitive level, as validating actions on the part level requires information about the tasks that are performed. Hence, two phases are distinguished in generating assembly process plans: part level planning and product level planning. The focus is on manufacturability, that is, finding a non-empty set of valid process plans. Selection of the optimal plan is of less concern. As manufacturability evaluation depends on the availability of a valid assembly sequence, plan representation and validation is addressed.

Part level planning

Part level planning concerns the mapping of part connections to assembly operations suited to realise these connections. The result is a set of assembly operations that is partially proven valid. The scope is limited to the connection that is of concern, without taking other parts or connections into consideration: a micro-process plan. When creating micro-process plans for assembly, the following needs to be considered:

- In the product model, a primitive completely describes a mate between two parts. Parts having no relation are not considered possible sub-assemblies, while the operations completely realise the mate between the parts.
- Whenever two parts are joined, all contacts are established and the parts remain connected. One may safely assume this is what the designer intended, so temporary connections are considered an assembly problem. As a result, plan representation will become less complex (unidirected).
- Research shows that in 90% of the cases, the volume of the parts does not exceed 400 cm^3 and their weight is below 2 kg. The number of parts is smaller than 25, while over 50% is assembled from one direction [Delc92].
- Reversibility is assumed, that is, it is possible to stop and reverse and action without changing the equipment. This rules out parts like springs and parts that are supposed to reach final position with the aid of gravity. This makes assembly planning by disassembly more generally applicable. Snap connections are a special case. They are not reversible, but can be disassembled using other operations (with access from the back side). As they are frequently used, a primitive may be created for a snap connection, but this is not elaborated.

The first concern is to find operations for joining two parts. A number of authors distinguish basic assembly operations [Delc92]. They are classified according to insertion movement (peg-in-hole, screw, force fit, bayonet joint), joining processes (crimp sheet metal, weld), handling operations (flip part, support), or combinations of specific parts (peg and retainer). These classifications are not systematic, mix up movements, processes, and connections, and vary in the amount of detail. Andreasen describes assembly as a compound of handling, composing, and checking [Andr88]. The handling process selects and prepares components for composing or checking. The composing process creates a permanent connection between components. The checking process checks a component's presence and position. All three processes are composed of storage operations, positioning operations, transporting operations, and special operations. The operations used when creating a connection are transporting and positioning operations, like moving, orientating, alignment, and insertion. Heemskerk uses four primary operations to describe the assembly of a part: feed, grasp, move, and mount [Heem90]. Although they were originally used in the planning process of a flexible assembly cell, sets of these operations can form micro-process plans for assembly. When planning the actual assembly activities, such as opening or closing grippers or moving robots, a test operation and a release operation can be added [Baar95]. For manufacturability assessment they are of less concern.

A combination of details from the complex primitives and the basic assembly operations is used here. Each of the primary operations is separately checked for validity. An operation requires specific equipment, tools, and checks. A micro-process plan is valid if all operations can be performed. Although this is equivalent to validity checking in parts manufacturing, a fundamental difference exists. Micro-process plan for assembly normally contain exactly the four operations mentioned and these operations remain together as a group. In parts manufacturing, the operations vary and are grouped differently when a manufacturing job is created, for instance according to tool or set-up (refer to Chapter 3). In a process plan for a milling machine for example, all centre-drilling operations in a set-up are grouped to minimise tool changes. In assembly, it is impossible to group for instance all move operations within a set-up. Because the result of an operation is input to another one, validity checking is performed in a certain order: feed, grasp, move, mount. To describe assembly actions or processes, a connection model is provided first.

Connection model

Insertion of a part takes place in three phases: approach motion, contact motion, and assembly motion. In the approach motion, a part is moved to the insertion point (including a coarse motion to get near the other part). In a contact motion, it is brought into contact with the assembly and pushed in the contact situation desired for the assembly motion, which moves it to the final location along an insertion path. The model is shown in Figure 5.1 [Mart91]. Relations with other parts of the assembly are discarded (part level planning). Assembly planning provides insertion points, approach direction sets, and insertion paths, while the operations move and mount realise the motions mentioned.

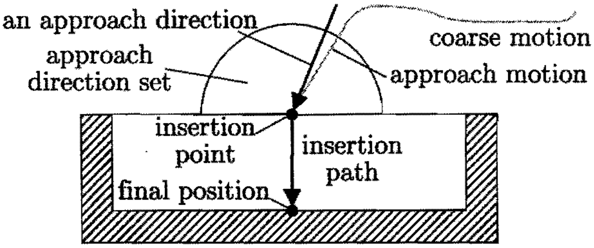


Figure 5.1: Connection model

In the classification by Andreasen, positioning a component concerns orientating it in relation to another component [Andr88]. This is performed by an alignment operation (positioning a component in one or more axial directions in relation to a base component) and an insertion operation (relocating a component to reach the terminal position), as shown in Figure 5.2. Gripping a component is considered part of the moving operation. In this research, alignment is a part of the fine motion performed in the mount operation, which is better suited for manufacturability evaluation.

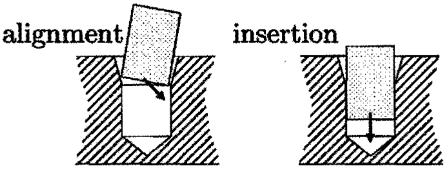


Figure 5.2: Positioning a component

Feed operation

In a feed operation, a loose part or sub-assembly receives a certain fixed location. It varies from transporting an assembly into the assembly station to the bulk feeding of parts. Planning activities concern selection of a feeder and a suitable location. A feeder is chosen mainly on the basis of economical considerations [Boot82]. The location of the part and the areas of the part that are occupied after the feed operation (faces a part rests on or that are blocked by the feeder) are crucial information to the grasp operation. These areas are not reference elements, but are part of the derived geometry (they are also partial faces). As the feed operation depends upon global part characteristics and economical considerations, it is not elaborated.

Grasp operation

In a grasp operation, a part is connected to a tool (gripper) to enable manipulation of its location. Planning activities are gripper selection, selection of positions of gripper contact areas on a part (a grip), and the motion of the gripper towards the part (a grasp). Selecting a gripper type is based on part material (elasticity, magnetic properties), weight, size, and shape. A selection is made between gripping based upon force closure (with special cases magnetism and vacuum) or form closure. Generally, the choice is between a gripper with two or three fingers, vacuum grippers, or magnetic grippers, as other grippers are more product specific. Apart from the type, decisions are made on details of a gripper, such as finger tips or suction cups. The compliance of soft tips for instance, may be advantageous or disadvantageous, while flat tips improve positioning, but have unclear contact properties. Selection of the set of suitable grippers is performed by query operations on a database of grippers. Generally, multiple grippers are found, of different types. This set is reduced as suitable grips are determined for each of the grippers.

Selecting a grip concerns finding the gripper contact areas available. A set of grips is found by assessing each of the faces of a part against three criteria: accessibility (no collision of gripper and part, faces at a maximum distance), vulnerability (threaded or polished faces), and stability (stable position, no slipping). For determining grips, the algorithm assumes the part to be in its final position to enable fast collision detection. In most cases, the type of gripper constrains the possibilities for the grasp (face normal direction required). From the list of faces, the faces are removed that are vulnerable, too far apart, or do not guarantee a stable grip. Collision detection is performed on the remaining faces by positioning the gripper volume near them. Faces that are used in the actual contact are not automatically ruled out. The parts of these faces that are not blocked by the contact are good candidates for gripping, as they posses a good orientation with respect to the assembly motion.

The stability of a grip, its tendency to return to original position when disturbed, is checked using some general rules. The faces in a grip need overlapping regions when projected to avoid unwanted rotations. The forces should be applied in the normal direction of a plane, while resistance to slipping depends on the normal force and the friction properties. The rules mentioned indicate possible problems. A more accurate description of the effects requires extensive calculations using models of contacts, friction, and so on. Such calculations can be found in for instance [Cutk85]. A grip also has a centre of compliance, where a force or torque only produces a deflection in the same direction. The centre of compliance is calculated from the information in a combination of gripper and grip. To prevent insertion problems, it is of importance that the contact point is close to the centre of compliance. Therefore, this information is stored for each grip and used during the mount operation.

Move operation

A move operation transports a part from a starting location to a location near the other part. The focus when validating a move operation is on collision detection. Four types of collision are identified, as depicted in Figure 5.3a through d:

- A location can be reached by the part, but the tool collides with the mating part: part level inaccessibility related to the tool.
- A location cannot be reached by the part due to the geometry of the mating part: part level inaccessibility related to mating part geometry.
- Part and tool do not collide with the mating part, but with another part in the assembly: product level inaccessibility. This generates a precedence constraint in sequence planning, but is not found here as only two parts are considered.
- A possible collision in the approach route. The complex route needed to reach a location is not found. This will not occur often, as parts possessing such odd geometry are difficult to manufacture, while manipulators suited for realising the complex approach motion without collision are hard to find. In some cases (manual assembly) the evaluator unjustly claims an error. However, this encourages designers to create products that are easier to assemble.

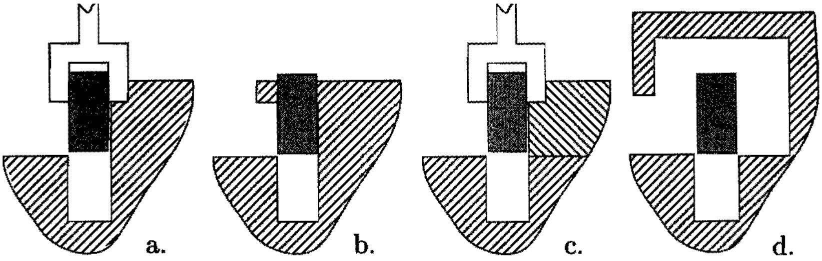


Figure 5.3: Collision in a move operation

The first check that is performed is not complex. For all grips, the combination of part and tool is positioned at the insertion location. To make the detection more realistic, part of the manipulator is modelled with the tool (such as the last joint of a robot). If a collision occurs, the designer is notified. Otherwise, an approach direction set has to be found. This set can be found by eliminating invalid directions from an initial set of directions by simulating product and tool movements. Experiments in collision detection performed to validate tool movements in parts manufacturing (simulation of numerically controlled machines) has proven that this is not a trivial problem. Simulation of moving solid objects, even when approximated by sets of discrete points, poses computational problems.

A solution is found in the (less computationally intensive) ray-test technique. A set of approach directions is seen as a sphere, as proposed in the obstruction concept [Mart91] and the visibility map [Woo90]. The region of the sphere indicating collision-free motions is approximated by firing rays from the part toward positions on the sphere. The diameter of the sphere determines the minimum length of the trajectory. If no other objects are hit, a collision free path exists. Creating this set based upon the insertion point is not sufficient, as Figure 5.4a demonstrates (two-dimensional representation of the problem). Therefore, all corners found on the part are used as a starting point for the test. Figure 5.4b depicts the collision free regions of three of the points. The minimal set of approach directions found guarantees a collision free path for the part. If the motion is reversed, B moves toward a instead of A towards B, the approach direction set is mirrored in the insertion point.

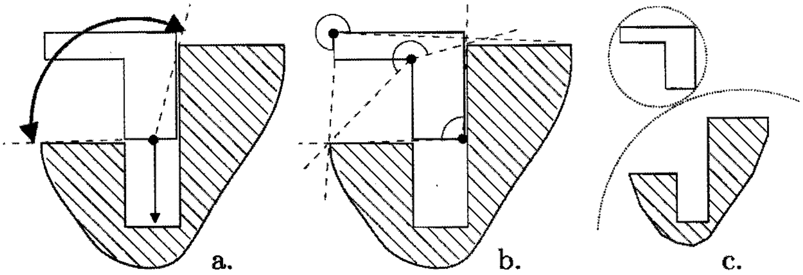


Figure 5.4: Collision detection

Making a distinction between approach motion (move) and an assembly motion (mount) makes sure the approach motion is not just the extension of the insert motion. As depicted in the connection model, the starting point for the mount operation is the insertion point. The starting location for the move operation in part level planning is found by an operation depicted in Figure 5.4c. The outer contours of the parts are described by spheres (bubbles). If the spheres do not overlap, the parts are totally free of each other (bubble check method). The part level move operation realises a trajectory from a valid starting point to the insertion point. The coarse motion that precedes the move operation is addressed in product level planning.

Mount operation

The mount operation realises the part relation. Moving a part to its final location is usually a motion with few degrees of freedom. In some cases, a complex movement is required to make the actual connection, but often it is only a fine straight line motion. The mount operation is determined by the properties of the connection, that is, it depends upon the restrictions described in the product model. The assembly direction, or at least a set of assembly directions, is determined by the assembly primitive. For most primitives, there are very few directions that are feasible.

A mount operation is a fine motion of a part that is (likely to be) in contact with its environment. Accurate modelling of such a motion requires information on dynamic behaviour and accuracy of equipment, contact forces, friction, and compliance. The mount operation has been fundamentally examined for a limited number of cases. Within the framework of mount operations, the round peg-in-hole is the single most frequently performed task (especially because of its circular symmetry). Still, the few theoretical models found tend to deviate from the data found in experiments. Besides that, assuming the mount operation is in itself reliable is not correct; implementations use controllers and sensors that are not part of the theoretical models.

Detailed models of the mount operation for assessment of its validity are therefore unreliable and impractical, especially if assessment is continuously performed during design. The more practical approach, based on the concept of comparing design requirements with capability, is therefore used for assembly as well (refer to Chapter 3). The mount operation assumes a nominal strategy, while the actual implementation uses a strategy capable of verification and recovery (controllers and sensors). The process capability is documented to enable manufacturability assessment. A graphical representation is depicted in Figure 5.5.

Capability of a mount operation is linked to a specific primitive (which is different from parts manufacturing). The mount operation for a peg-in-hole primitive for instance, is different from the mount operation for a peg-in-hole primitive with a chamfer or a square peg-in-hole. The process capability depends on compliance and position accuracy of the equipment. For each primitive, these properties are compared with the clearance that is available and a characteristic length, usually the length of the contact. If no equipment with the required properties is available, a manufacturability problem exists. As in parts manufacturing, the capability depends

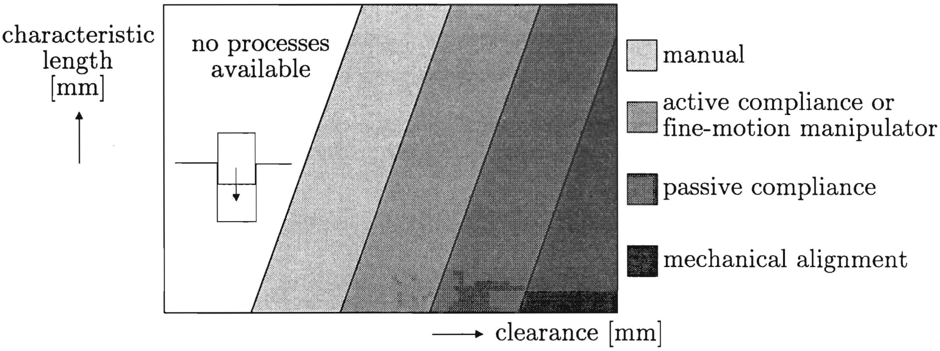


Figure 5.5: Assembly capability for a peg-in-hole primitive

upon the production system involved. If a factory specialises in accurate manipulators, the range of achievable clearances will be different from a factory using standard robot assembly cells. Permitting manual assembly reduces the number of problems, as the achievable accuracy and compliant motion of the human hand is generally unequalled. Quantitative data is not provided in Figure 5.5, as it is not based upon actual experiments in manufacturing systems (as opposed to Figure 3.8). Figure 5.5 mentions some possibilities for motion and contact:

- Mechanical alignment tries to create equipment that is able to accurately follow programmed paths. Small variations result in mechanical contacts without feedback (stiffness). This approach is relatively expensive, errors may have severe consequences and it is not suited for small clearance values.
- Force feedback creates a relation between forces and displacements, reducing the risk of damage. It can be implemented as a mechanical system (passive feedback) or extended with a control loop (active feedback).
- A coarse motion and fine manipulation using a combination of different manipulators, resulting in better dynamical behaviour, but higher complexity. General purpose manipulations cannot achieve both large manipulation and fine-motion manipulation (like simulating a human arm, wrist, and hand).

The relative location of the parts is important. Parts are aligned and positioned by the assembly equipment, while clearance, compliance, and the primitive determine if the accuracy of this equipment is sufficient. The absolute location of one part is manipulated, the other part serves as a base part. The uncertainty in the location of the base part is not addressed. This is compensated by lowering the width of the envelopes (the result of the tolerance propagation) of the relevant faces in the connection. This decreases the clearance available, so if the base part cannot be located accurately, more manufacturability problems occur in mounting.

Product level planning

Where part level planning considered contacts between two geometrical objects, product level planning is concerned with the whole product and tries to find a suitable assembly sequence. Generating an assembly sequence consists of finding precedence relations, feasible assembly sequences, and an optimal sequence. Generating an assembly sequence is a combinatorial process, the complexity increases exponentially with the number of parts involved. A product with 10 parts generates millions of combinations (n parts generate n! combinations).

Representation

The choice of the representation of assembly sequence is an important decision. It needs to handle a large amount of combinations, while searching for feasible sequences should be possible in a reasonable time period. Consequently, combinations that are known to be infeasible are not added (like sub-assemblies of parts having no connections), while evaluation results are used to eliminate groups of invalid combinations as soon as possible (generate a tree as it is searched and do not expand invalid nodes). A survey of models, representations of sequence, and solving strategies is found in [Delc92] and [Heem90]. Representations are for instance triangle tables, precedence diagrams, AND/OR graphs, temporal logic, and Petri Nets.

As an example, Figure 5.6 depicts the AND/OR graph of a product consisting of four parts (A, B, C, D). The nodes corresponding to feasible sub-assemblies are found by identifying possible disassembly movements [Home91], so the graph is created from the top down. Product $ABCD$ is disassembled into (ABC AND D) OR (BCD AND A). Other combinations are not possible as B and C cannot be removed from the assembly. Sub-assemblies ABC and BCD are disassembled into valid sub-assemblies AB, AC, BC, BD, and CD. These are disassembled to create single parts. Figure 5.6 shows that this is not an orderly representation. The connections between the nodes are often hard to follow as they cross the other connections and nodes.

Here, a representation is chosen that fits the needs and corresponds best with the terminology used. For representing the assembly process, Heemskerk presented a method using states and transitions [Heem90]. Admissible transitions are the ones

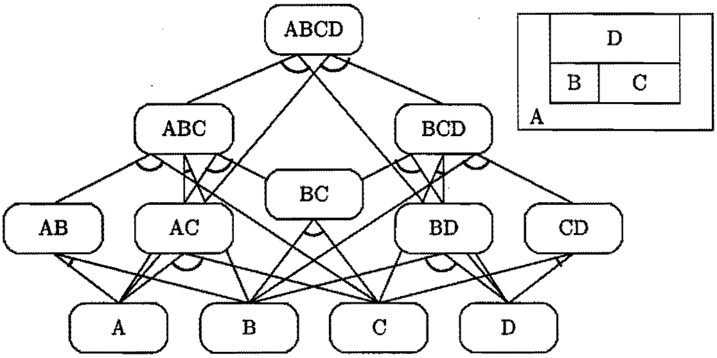


Figure 5.6: AND/OR graph representation

that are accessible and stable. This matches the concepts presented in this research. Accessibility and stability generate precedence relations and thus restrict the number of valid sequences. If no sequence is found, an assembly problem exists. The assembly states and transitions are represented in an Assembly State Transition Diagram (ASTD). Figure 5.7 shows the ASTD of the example product (parts A, B, C, D).

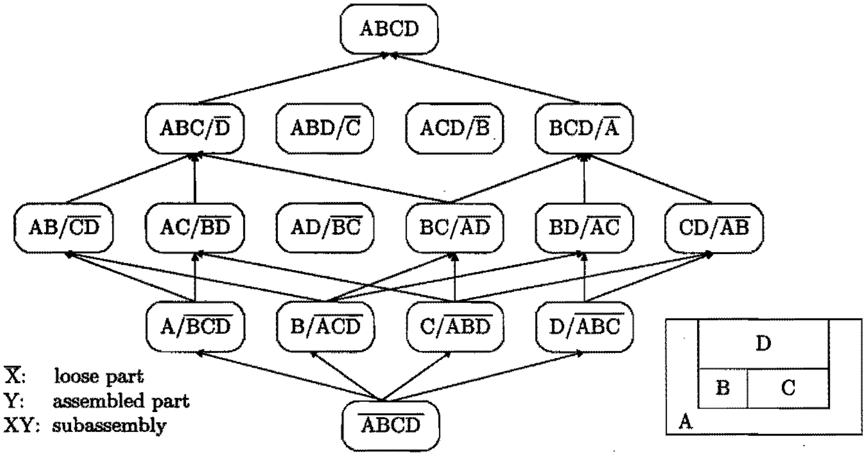


Figure 5.7: Assembly state transition diagram

The ASTD is a directed graph that indicates which parts are loose (notation: $\overline{X}$), which parts are assembled (notation: Y), the sub-assemblies (notation: XY), and all admissible assembly sequences. If multiple sub-assemblies ABC and PQR are present, this is described as $ABC/PQR/\overline{XYZ}$ for instance. The creation of an ASTD starts at the bottom. The goal state contains the product, while the start state contains all loose parts. During a transition, a part or sub-assembly is assembled. A path from start state to goal state corresponds to an assembly sequence. As demonstrated in Figure 5.7, some states are irrelevant, as they are not used in any valid sequence (states $ABD/\overline{C}$ and $ACD/\overline{B}$, in which part D blocks C or B).

Joining sub-assemblies, a state transition, is an assembly task. An assembly sequence is valid if all assembly tasks are feasible. A task consist of the operations mentioned: feed grasp, move, mount. This representation is a clear manufacturing representation, as a concept like connections (which are design elements) does not exist.

Reducing complexity

A number of procedures enable complexity reduction of the process planning:

- To reduce the number of nodes in an ASTD, the number of sub-assemblies may be limited, a base part can be chosen, or parts can be grouped into clusters (which results into a layered representation, the LASTD).
- If the number of checks that has to be performed is too large, an option is to select a limited number of trajectories that are more likely than others.

A base part is a part having a lot of (or most) connections, a part that is relatively difficult to manipulate (large, heavy), or a part that enables assembly of all other parts from one direction (a frame). Besides reducing the number of nodes, selecting a base part will make fixture planning easier. In some cases, there may be a functional reason for selecting a base part.

Selecting a limited number of trajectories that are likely provides another reduction in complexity. Such trajectories are the ones perpendicular to plane contact surfaces, the axis of cylindrical surfaces, or extensions of the insertion direction. The information for this selection is provided in the primitives. Other likely, but more complex trajectories can be derived from the complex primitives as well (like for fasteners). It is also possible to restrict the assembly trajectories to a limited set of directions (like only assembling from above). Most of the trajectories mentioned however, have limited scope and are more useful for part level planning. Besides that, they risk not finding valid plans that do exist. If plans are found using these simplifications however, they tend to have a greater ease of assembly. The procedure for finding feasible assembly sequences then contains four steps:

1. select a base part
2. build the ASTD while validating the assembly tasks for the new nodes, invalid nodes are not expanded further
3. if no sequence is found, a new base part may be selected
4. if needed, an optimal solution is chosen regarding some criterion

The example provided in Figure 5.7 then reduces to a problem with 6 nodes, compared to 16 before. Selecting A as a base part renders a state transition from $\overline{ABCD}$ to $A/\overline{BCD}$. The next step is $AB/\overline{CD}$ or $AC/\overline{BD}$, both resulting into $ABC/\overline{D}$ and finally $ABCD$ (the left most branch of the diagram in Figure 5.7).

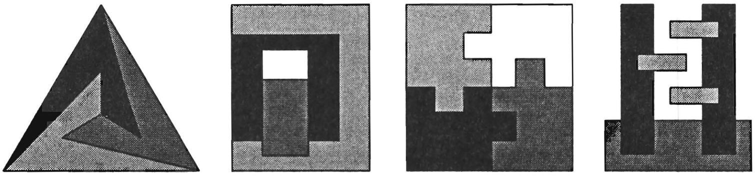
Creating an ASTD is basically a form of forward planning, a path from loose parts to the assembled product. One would expect that finding disassembly sequences (backward planning) is easier, as every step leads to a smaller product, while assembling leads to a bigger product with more chances for obstruction. However, in backward planning, backtracking will occur, especially in optimisation. Besides that, backwards planning suffers from the assembly partitioning problem: parts or sub-assemblies need to be selected, without sufficient information to make a choice. Selecting a base part in forward planning is not complex.

Problems

When generating an assembly sequence, a number of problems can be identified [Wolt88]. They are found in product level planning, as they only occur when more than two parts are of concern. Each of them requires a specific type of process plan. Two-dimensional examples are depicted in Figure 5.8a through d:

- A non-sequential assembly plan: more than one part moves at a time (a).
- A non-monotone assembly plan: parts have temporary positions (b).
- A non-linear assembly plan: not possible without creating sub-assemblies (c).
- A non-coherent assembly plan: sub-assemblies use parts with no connection (d).

Non-monotone and non-coherent assembly plans (b. and d.) do not meet the assumptions made at the beginning of this chapter, as parts are put in their final



a. b. c. d.

Figure 5.8: Problems in assembly planning

position and sub-assemblies need to have a connection. Non-sequential assembly plans (a.) describe a timing problem. Although this was not specifically addressed before, the algorithms presented are unable to detect or solve this type of problem. Non-linear assembly plans (c.) can be solved, but this requires validating nodes with more than one sub-assembly, which dramatically increases complexity.

Non-monotone plans are generally disregarded, because they are uncommon and hard to create (the graphs described above can only represent monotone plans). As most assembly stations only have one manipulator, while the use of multiple manipulators working on one product at the same time causes synchronisation problems, assuming sequential plans is acceptable. Assemblies that are monotone but non-linear are more common but most planners are unable to deal with them due to complexity. If one sub-assembly is accepted in the assembly state, most of the problems are solved however. For a collection of parts $ABCDEF$, state $AB/CD/\overline{EF}$ (assembled parts A and B, sub-assembly CD, loose parts E and F) is accepted, while $A/BC/DE/\overline{F}$ (two sub-assemblies BC and DE) is not accepted. Despite this constraint, the example of 5.8c can be solved. Products requiring a non-coherent assembly plan are a problem, because in most cases the information needed is not present. If algorithms are used that require a representation of the contacts, this prevents any non-coherent plan from being built. Only representations that do not use this coherence constraint are able to deal with the problem. The approach used here is not of is this type.

Non-sequential and non-monotone plans are disregarded here. Because of complexity and a lack of information, the designer (process planner) is relied upon to provide hints in case of non-linear and non-coherent plans. Because sub-assemblies, especially non-coherent sub-assemblies, often require special handling, the designer is not encouraged to create parts requiring these types of plans. Therefore, they are initially regarded an assembly problem. Only if the designer (process planner) specifically indicates that a sub-assembly should be created, the parts are treated as such. This results in a layered ASTD, where a sub-assembly is included in the (top) ASTD as one part. The sub-assembly has its own ASTD, which comprises another (lower) layer.

Optimisation

The assembly sequence drastically effects the efficiency of the assembly process. From a number of feasible sequences, an optimum could be chosen. Although it is not the primary target of the assembly sequence planner in this research, an indication is provided on optimisation criteria:

- directionality; insertion from a single direction is preferred
- fixture complexity; partial assemblies hold themselves together
- manipulability; perform difficult operations with parts that are easy to handle
- uniformity; assemble similar parts in similar ways to reduce number of tools
- locality; perform operations that are near each other consecutively so save time
- accessibility; make a sequence that results in as much clearance as possible
- tool changes; operations with same or similar tools together to save time
- parallelism; try to perform operations at the same time, use sub-assemblies
- production planning aspects; due to availability of workstations, fixtures or tools, certain plans are preferably avoided

These criteria may contradict (use of sub-assemblies) or require very extensive computations to optimise (uniformity, accessibility). Besides that, they depend upon the assembly method involved (manual, assembly cell, assembly line) and the relative weight is unclear. The locality criterion discards the move and grasp operation. Due to the fast feedback introduced in this research, a designer will have the tendency to change part geometry instead of trying to find sub-assemblies, complex motions, or fixtures. As this benefits the ease of assembly, somewhat limited, non-optimal algorithms for validating assembly operations are not entirely disadvantageous.

Validation

In product level planning, it is known that combinations of two parts can be assembled. It is still unclear however, if there is a collision with other parts, both in the approach motion and the coarse motion. For finding a valid path, techniques used in robot motion planning can be used, such as the Configuration Space (C-space) formulation [Lato91]. The idea is to represent an object as a point in an appropriate space, the configuration space. This transforms the problem of planning the motion of a dimensioned object into the problem of planning the motion of a point. Planning approaches in C-space are road-map methods (using for instance visibility graphs or Voronoi diagrams [Held91] to compute road-maps), cell decomposition (decomposing free space into simple regions), and potential fields (represent a point as a particle moving in a potential field). For coarse motion planning and finding collision free paths, these methods are preferred over simulating the movement of parts along trajectories (solid modelling operations). This simulation suffers from the problems mentioned in part level planning. Due to the increase in complexity in product level planning, these problems are almost impossible to solve.

Stability is checked by comparing the remaining degrees of freedom in a connection of two parts to the direction of gravitational forces and insertion forces. Although useful in some cases, friction is not taken into account at this point, so sub-assemblies relying on friction to remain stable are considered an assembly problem.

5.2 Inspection operations

The assembly operations assume the parts to be within the specifications. The envelopes resulting from the specified tolerances are reasoned with in assembly, while the inspection operations preceding assembly are disregarded. Ensuring the functionality of assemblies however, involves measuring the parts to verify that the dimensions are within designer-specified tolerances. This makes inspection planning an integral part of manufacturing planning.

A (face) reference element that is part of a tolerance relation is measured by a co-ordinate measuring machine. Because the envelope is part of the reference element, the measurements can be compared with it for verification. As reference elements have infinite dimensions in some directions, the boundaries of a face are determined from the geometrical model. Analogous to for instance numerical control code for milling machines, an interface to co-ordinate measuring machines is needed. The Dimensional Measuring Interface Specification (DMIS) is a suitable standard for the bi-directional communication of inspection data between computer systems and inspection equipment [Sars95, Scho95]. It is a result of the CAM-I Quality Assurance Program (ANSI/CAM-I 101-1995). DMIS provides primitives such as a cylinder and a planar face, along with tolerance calculations regarding these primitives.

Because inspection is an integral part of manufacturing planning, manufacturability evaluation is extended with criteria concerning measuring. During design, design transformations containing strict tolerances are checked for a suitable distribution of measuring points and measuring probe access.

5.3 Manufacturing grammar

The concepts presented in the design grammar (Chapter 4) have a manufacturing counterpart, as depicted in Figure 5.9. For manufacturing, a set of symbols and productions can be identified as well. The symbols used in design are mapped upon the manufacturing symbols. This is performed at two levels. The design substates and assembly transformations are transformed into parts and assembly tasks. At a lower level, part manufacturing operations are found to create the parts.

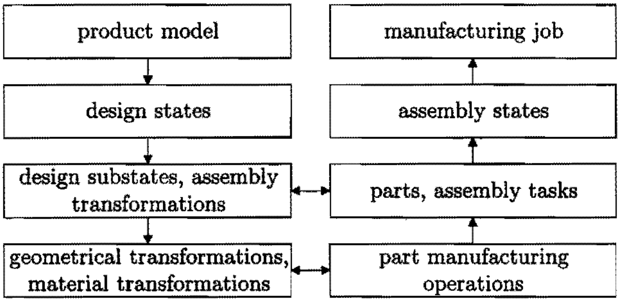


Figure 5.9: Design and manufacturing

An assembly state was defined as a valid collection of loose parts, assembled parts (the assembly), and sub-assemblies, while an assembly task is a collection of assembly operations. In manufacturing, a part is described by the manufacturing operations needed to create it (part manufacturing plan). Parts and sub-assemblies are joined using assembly tasks. A valid order of assembly states (and therefore assembly tasks) is described in the assembly sequence. Generally, multiple sequences are possible (the optimal manufacturing job will contain a single sequence). The symbols of the manufacturing grammar are described below (EBNF notation).

$$\begin{aligned}
\langle Manufacturing\ Job\rangle ::&= \{\langle Assembly\ Sequence\rangle\} \\
\langle Assembly\ Sequence\rangle ::&= \{\langle Assembly\ State\rangle\} \\
\langle Assembly\ State\rangle ::&= \{\langle Part\rangle\}\ [\langle Assembly\rangle]\ [\langle Sub-assembly\rangle] \\
\langle Part\rangle ::&= \langle Part\ Manufacturing\ Plan\rangle \\
\langle Assembly\rangle ::&= \{\langle Part\rangle\}\ [\{\langle Sub-assembly\rangle\}]\ \{\langle Assembly\ Task\rangle\} \\
\langle Sub-assembly\rangle ::&= \{(\langle Part\rangle\ \langle Assembly\ Task\rangle)\} \\
\langle Assembly\ Task\rangle ::&= \langle Feed\ Operation\rangle\ \langle Grasp\ Operation\rangle \\
&\quad \langle Move\ Operation\rangle\ \langle Mount\ Operation\rangle
\end{aligned}$$

$$\begin{aligned}
\langle Feed\ Operation\rangle ::\ &=\ \langle Feeder\rangle\ \langle Location\rangle\ \{\langle Non-free\ Region\rangle\}\\
\langle Grasp\ Operation\rangle ::\ &=\ \{(\langle Gripper\rangle\ \{\langle Grip\rangle\})\}\\
\langle Move\ Operation\rangle ::\ &=\ \langle Start\ Location\rangle\ \langle Insertion\ Point\rangle\ \{\langle Trajectory\rangle\}\\
\langle Mount\ Operation\rangle ::\ &=\ \langle Insertion\ Point\rangle\ \langle Insertion\ Path\rangle\\
\langle Gripper\rangle ::\ &=\ \langle Gripper\ Type\rangle\ \langle Gripper\ Geometry\rangle\ \langle Reach\rangle\\
\langle Grip\rangle ::\ &=\ \langle Position\rangle\ [\langle Position\rangle]\ [\langle Position\rangle]\ \langle Compliance\rangle
\end{aligned}$$

The right hand symbols in this notation that are not expanded further, such as location, non-free region, gripper geometry, point, or trajectory, exist as entities in the implementation. They can be expressed as geometrical entities (in a solid modeller) or as identifiers (like a string expressing a type of machine). The compliance is a stiffness matrix, depending on gripper properties and the contact positions of the gripper on the part. It expresses the movement as a function of the applied force for each degree of freedom. The number of contact positions used in a grip depends on the gripper type.

The productions deriving the strings of the grammar of manufacturing are analogous to the ones described in the design grammar. Symbols are expanded until the right hand side of the string consists of primitives. The manufacturing grammar is illustrated here first. After that, part of the mapping of the design operations upon manufacturing operations is described. The example that was already used in Chapter 4 serves as a guide, as depicted in Figure 5.10.

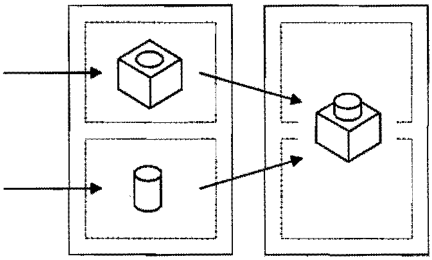


Figure 5.10: Product manufacturing example

For creating this product, one assembly sequence is applicable (the block is the base part). This sequence contains two states, a collection of two loose parts and an assembly of these parts.

$$\begin{aligned}
\langle Manufacturing\ Job\rangle\ &\rightarrow\ \langle Assembly\ Sequence\rangle\\
\langle Assembly\ Sequence\rangle\ &\rightarrow\ \langle Assembly\ State\rangle\ \langle Assembly\ State\rangle\\
&\rightarrow\ \langle Part\rangle\ \langle Part\rangle\ \langle Assembly\rangle
\end{aligned}$$

Each of the parts has a part manufacturing plan, creating the block with a hole and the peg respectively. These part manufacturing plans are derived from the descriptions of the parts, which will not be elaborated here. A thorough examination of the contents of such plans can be found in [Vrie96]. The assembly consists of two parts and the four operations used to connect them. For creating the contents of these operations, detailed information from the product model is needed.

$$\begin{array}{rcl} \langle \mathit{Assembly} \rangle & \rightarrow & \langle \mathit{Part} \rangle \, \langle \mathit{Part} \rangle \, \{ \langle \mathit{Assembly\ Task} \rangle \} \\ & \rightarrow & \langle \mathit{Part} \rangle \, \langle \mathit{Part} \rangle \, \langle \mathit{Feed\ Operation} \rangle \, \langle \mathit{Grasp\ Operation} \rangle \\ & & \langle \mathit{Move\ Operation} \rangle \, \langle \mathit{Mount\ Operation} \rangle \end{array}$$

The move operation and mount operation are used to demonstrate the mapping. Information is needed concerning the start location, the insertion point, the trajectory, and the insertion path.

$$\begin{array}{rcl} \langle \mathit{Move\ Operation} \rangle & \rightarrow & \langle \mathit{Start\ Location} \rangle \, \langle \mathit{Insertion\ Point} \rangle \, \langle \mathit{Trajectory} \rangle \\ \langle \mathit{Mount\ Operation} \rangle & \rightarrow & \langle \mathit{Insertion\ Point} \rangle \, \langle \mathit{Insertion\ Path} \rangle \end{array}$$

In the design grammar, the assembly information was presented as:

$$\begin{array}{rcl} \langle \mathit{Assembly\ Transformation} \rangle & \rightarrow & \mathit{Peg\ in\ hole} \\ & & \langle \mathit{Reference\ Element} \rangle \circledcirc \langle \mathit{Reference\ Element} \rangle \\ & & \langle \mathit{Reference\ Element} \rangle \veebar \langle \mathit{Reference\ Element} \rangle \end{array}$$

The reference elements in these assembly relations are described in the parts. Using the information in these parts, such as the location of the cylinder in the part and its envelope, renders quantitative information. In the product model, the information was presented as:

$$\begin{array}{rcl} \langle \mathit{Relation} \rangle & \rightarrow & \mathit{Plane\ False}\ (1,0,0)\ (100,0,0)\ \langle \mathit{Envelope} \rangle\ //\ 50 \\ & & \mathit{Cylinder\ False}\ (0,1,0)\ (50,0,0)\ \langle \mathit{Envelope} \rangle \\ \langle \mathit{Envelope} \rangle & \rightarrow & (0,1,0)\ (50,0,0)\ 0.1 \end{array}$$

From the geometrical informations in the part and the assembly relations, it can be concluded that the insertion point is (50,50,100). The trajectory for the move operation is in half a sphere, limited by a plane with normal vector (0,0,1) and vertex (50,50,100). The insertion path is a straight line with direction (0,0,-1). From the envelopes of the peg and the hole it is concluded that mechanical alignment suffices for the mounting operation.

5.4 Summary

In this chapter, the characteristics and terminology of assembly process planning were assessed. Validity checking of assembly operations was performed and design operations were mapped upon manufacturing operations. Two phases are distinguished in assembly process planning: part level planning and product level planning.

A part is assembled using the operations feed, grasp, move, and mount. In part level planning, each of these operations is separately checked for validity to make sure an assembly is manufacturable. The focus when validating the operations is on collision detection and accuracy. Detailed models of the mount operation (realising the part relation) for assessment of its validity are unreliable and impractical. A more practical approach is based on the concept of comparing design requirements with capability, analogous to part manufacturing assessment.

Product level planning tries to find a suitable assembly sequence. A representation of assembly sequence is chosen that fits the needs and corresponds best with the terminology used. This method uses states and transitions represented in an Assembly State Transition Diagram. Besides the need for reducing complexity, some problems exist that require specific types of assembly plans: non-sequential, non-monotone, non-linear, and non-coherent assembly plans. In some cases the designer or process planner needs to provide hints for a solution. As the assembly sequence drastically effects the efficiency of the assembly process, an indication was provided on optimisation criteria.

For finding a valid plan in product level planning, techniques used in robot motion planning are used, while stability is checked by comparing degrees of freedom to the direction of gravitational and assembly forces. Besides that, an indication was given on the incorporation of inspection planning, which is an integral part of manufacturing planning.

As in design, the concepts used in manufacturing planning can be expressed in a grammar. The quantitative details in the strings of this manufacturing grammar are found by interpreting the design grammar.

Chapter 6

A case

This chapter demonstrates the use of the concepts presented in the foregoing chapters. Following an introduction to the tool, design and process planning is performed for an example product, derived from an actual application in industry.

6.1 A tool

Implementation of a design support tool is in the form of *proof of concept software*; incomplete, but suited for creating designs. Two types of implementation were considered: extension of an existing CAD system and creating a new tool. In a CAD system, one is forced to comply with the structure and entities, while product model data is often inaccessible. Advantages are the availability of a user interface and geometric modelling tools. Creating a new tool does not mean it is built from scratch however, as software packages for parts of the CAD functionality are commercially available, while links to existing databases are possible. Still, creating a working prototype requires a lot of programming effort [Net95, Vrie96]. Considering the problems encountered in extending CAD systems, the tool is implemented by programming an application (Visual C++) that communicates with a solid modeller (ACIS) through an application procedural interface (API). A user interface enables product model definition, propagation, process planning, and feedback.

6.2 Creating the product

The product chosen concerns a welding module used in production lines for mass production of consumer goods. It joins wires or plates by resistance welding. It is a low volume, engineer to order product, created in a job shop production environment. The parts do not have a complex geometry, but the quality demands are high and a short lead time is required.

Product creation process

In a traditional product creation process, a conceptual design of such a module is created by an engineer that is familiar with its functions and the relations it has to the product and other components of the line. The conceptual design is provided with the most important dimensions and passed on for detailing. Tolerances are added, based upon company standards and the designers experience. Manufacturing planning is performed through interpretation of the geometrical model or drawings, with extensive support from the human process planner. Errors that appear in this stage are reported to the person responsible for the details. After assembly, the functionality of the product is assessed by the engineer that was responsible for the conceptual design. If problems originate from erroneous detail decisions, this is reported to the detailer, thus increasing experience.

Conceptual design

The welding module joins wires or plates with wire diameters or plate thickness from 1.2 to 2 millimetres. This requires a voltage of 0-6 Volts, a current up to 1500 Ampere, and a force up to 50 Newton. Adapting to specific products is done by changing the voltage, current, force, welding time, and electrodes. Figure 6.1 shows a conceptual design. An air cylinder realises a controlled vertical stroke, the force is applied by a spring. The maximum dimensions are 450×200×200 millimetres, the stroke must be at least 25 millimetres, and power and air must be supplied.

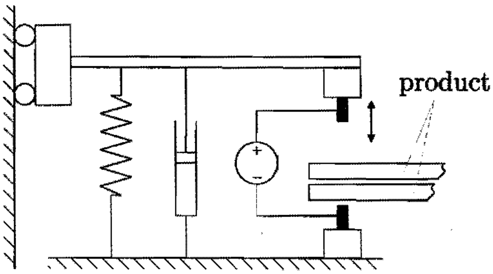


Figure 6.1: Conceptual design of a welding module

The design session described below is based upon the conceptual design. The limitations of the design tool however, drastically influence the product design itself. As standard parts like bearings and guideways are not available in the tool, they are replaced by parts created by the designer. These parts roughly perform the same function, but achieving equal performance is very difficult. As a result, the product design that results from a design session with the current tool is different from a design created in other tools or using drawings. The purpose of the design session however, is to clarify the concepts proposed. For convenience, the fact that the product may not even function properly due to the limitations is disregarded here.

The validation and process planning of the design also partially relies on imaginary aspects of a support tool, as some of the algorithms are marginally implemented. The creation and propagation of relations and process planning for parts is fully operational. Validation of grips and approaches and assembly sequence planning are still in the conceptual phase.

6.3 Designing the assembly

Figure 6.2 depicts the parts of the welding module except for the cylinder and the spring (parts that are from outside suppliers). A base plate (A) connects the module to the rest of the production line and supports the other parts. A support structure (B), a guiding plate (C), and a moving holder for the upper electrode (D, E, F) realise the vertical stroke. The lower holder for the electrode (G) is mounted on the base plate. The following sections describe how the parts were created and assembled.

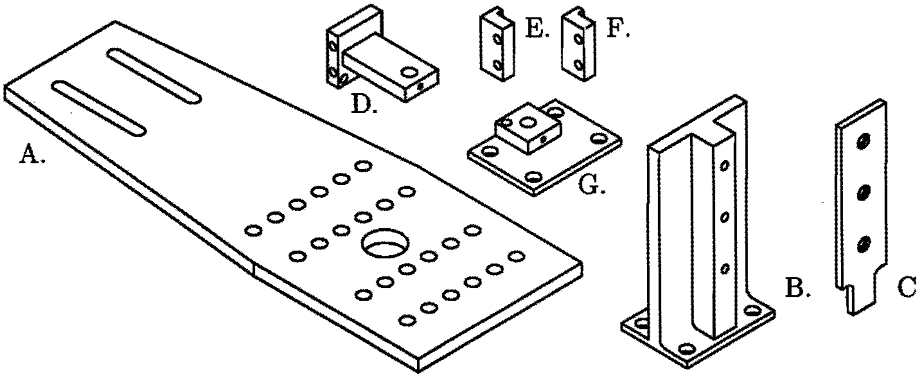
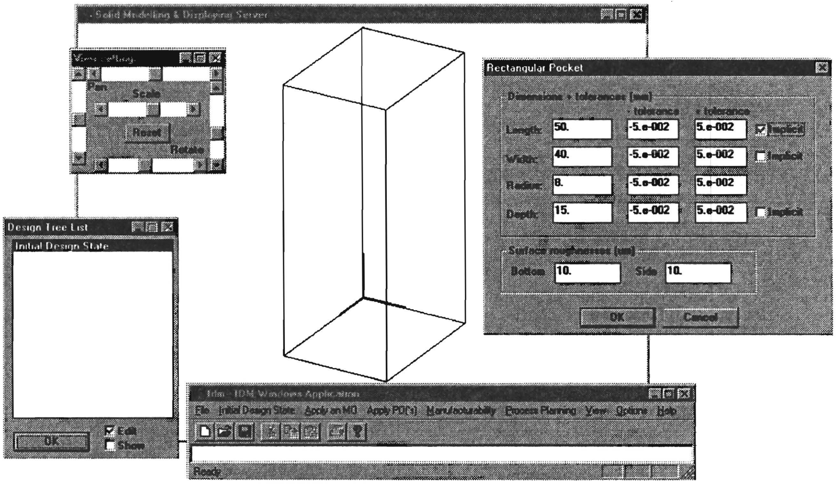


Figure 6.2: The parts of the welding module

Product modelling

The geometrical design of the support structure (part B) is used to demonstrate part modelling in the design support tool. When the part is created, a new design substate is introduced in the product model. By selecting a primitive and a material, an initial design state for the new part is created. The selection of the primitives determines what reference elements are available to start with. Here a box primitive is selected. The design transformations that follow are kept in the design tree list. The geometry that resulted from the transformations in this list is displayed.

Following the definition of an initial state, the designer selects primitives and operators to specify design transformations. First, a pocket primitive is specified to remove a piece of material, as depicted in Figure 6.3 [Net95]. The designer is provided with default dimensions and tolerances that can be altered and an option to indicate that they are described using relations (implicit). In view of the concepts presented in this research, implicit definition of dimensions and location is preferred.



Dimensions + tolerances [mm]

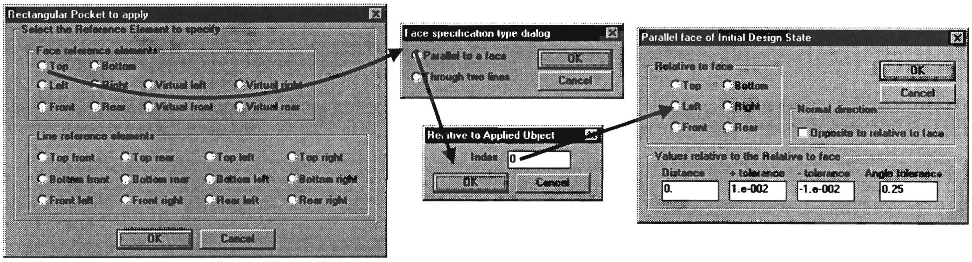
		- tolerance	+ tolerance	
Length:	50.	-5.e-002	5.e-002	☑ Implicit
Width:	40.	-5.e-002	5.e-002	☐ Implicit
Radius:	8.	-5.e-002	5.e-002	
Depth:	15.	-5.e-002	5.e-002	☐ Implicit

Surface roughnesses [um]

Bottom: 10.
Side: 10.

Figure 6.3: Creating a rectangular pocket primitive

The pocket specified is used to remove material, so tolerances are of less concern and virtual reference elements may be used. The relations are created through a series of dialogues, as depicted in Figure 6.4. The available reference elements are presented (depending on the primitive). When an element is selected, the options available for locating it are provided, and a choice must be made for the other reference element. Tolerance relations are not separate relations, but are added to the geometrical relations (based upon considerations presented in Chapter 4). As long as the primitive is not completely specified, new relations must be added.



Index: 0

Distance	+ tolerance	- tolerance	Angle tolerance
0.	1.e-002	-1.e-002	0.25

Figure 6.4: Dialogues in specification of relations

The relations that are used are presented in Table 6.1. The only parameter entered here is the corner radius (7 millimetres). As the radius has no functional use, entering it as a parameter is easier than specifying another relation.

	Pocket reference element	Box reference element	Operator and dimension [mm]
1.	top	left	// 0
2.	bottom	left	// 25
3.	front	bottom	// 5
4.	rear	top	// 0
5.	virtual left	rear	// 0
6.	virtual right	front	// 0

Table 6.1: Relation describing a pocket

Figure 6.5 depicts the primitive and the reference elements specified after each relation (propagation). The first relation determines the top face of the primitive (1), and thus restricts two rotations of the primitive and the vertical direction of the position. When the bottom reference element is added (2), the depth of the primitive is specified. The front reference element (3) restricts the third rotation and determines the longitudinal direction of the position. The rear reference element (4) determines the length. This leaves the dimension and position in lateral direction. These are determined by the virtual left and virtual right reference elements (5 and 6). The feedback is depicted in Figure 6.6. The result of the propagation is displayed, along with the information about the micro process plan.

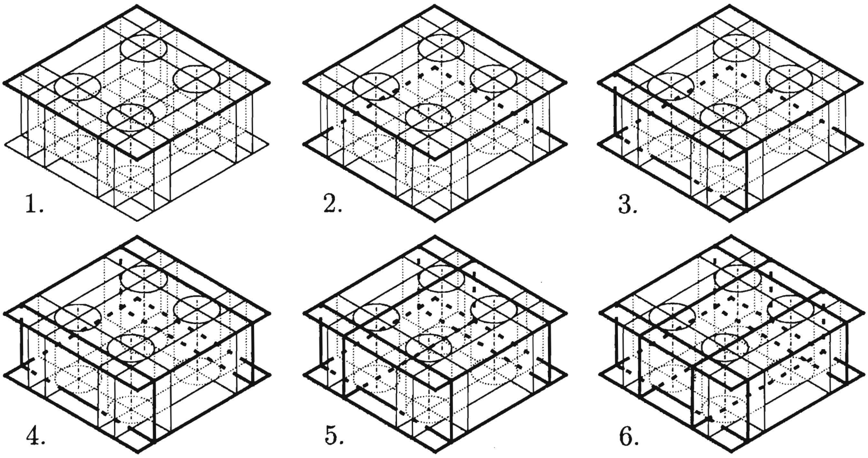
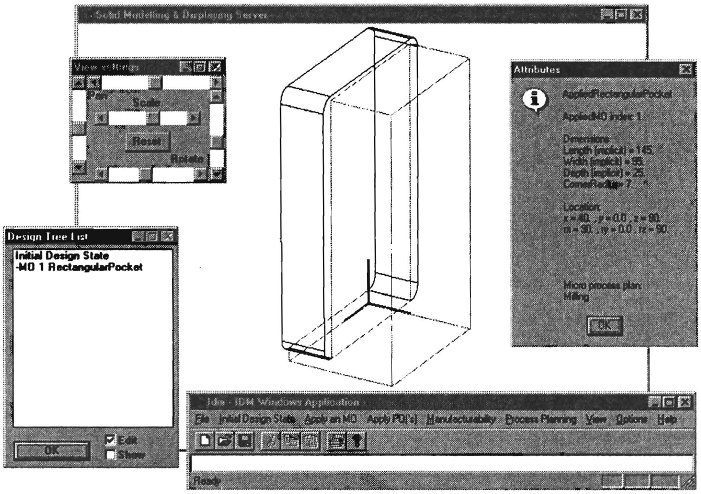


Figure 6.5: Propagation of rectangular pocket relations



Initial Design State
-MO 1 RectangularPocket

AppliedRectangularPocket

AppliedMO index 1

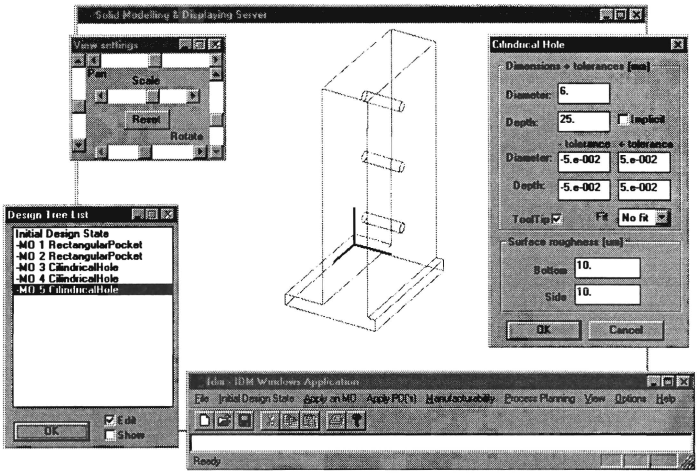
Dimensions:
Length (implicit) = 145.
Width (implicit) = 55.
Depth (implicit) = 25.
CornerRadius = 7.

Location:
x = 40. , y = 0.0 , z = 80.
rx = 90. , ry = 0.0 , rz = 90.

Micro process plan:
Milling

Figure 6.6: Result of applying the primitive

To attach the guiding plate (C), the base is provided with holes through cylinder primitives, as depicted in Figure 6.7. The dialogues used in the definition are presented in Figure 6.8. The centre line is parallel to two faces of the initial design state and the top of the cylinder coincides with the bottom of a pocket. As the depth is entered as a parameter, this completely defines the cylinder primitive.



Design Tree List:
- Initial Design State
- -MO 1 RectangularPocket
- -MO 2 RectangularPocket
- -MO 3 CilindricalHole
- -MO 4 CilindricalHole
- -MO 5 CilindricalHole

☑ Edit
☐ Show

Dimensions + tolerances [mm]

Diameter: 6.

Depth: 25. ☐ Implicit

	- tolerance	+ tolerance
Diameter:	-5.e-002	5.e-002
Depth:	-5.e-002	5.e-002

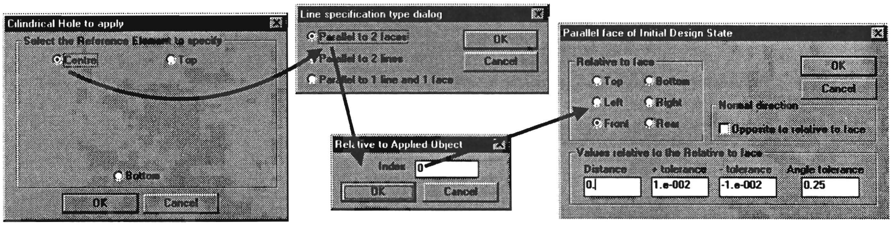
ToolTip ☑ Fit: No fit

Surface roughness [um]

Bottom: 10.

Side: 10.

Figure 6.7: Creating a cylinder primitive



Select the Reference Element to specify
● Centre ○ Top ○ Bottom

● Parallel to 2 faces
○ Parallel to 2 lines
○ Parallel to 1 line and 1 face

Index: 0

Relative to face
○ Top ○ Bottom
○ Left ○ Right
● Front ○ Rear

Normal direction
☐ Opposite to relative to face

Values relative to the Relative to face

Distance	+ tolerance	- tolerance	Angle tolerance
0.	1.e-002	-1.e-002	0.25

Figure 6.8: Dialogues in specification of relations

Besides geometrical transformations, assembly transformations are specified to connect the parts. Although it is possible to mix geometrical transformations and design transformations, that is, to assemble parts before they are finished, this property is not used here. It is assumed that all part are finished. In practical applications, the mixing off geometrical transformations and assembly transformations will be quite common, as a lot of geometrical details are derived from the assembly requirements. Figure 6.9 shows the positions of the parts in the final assembly (details are left out).

For demonstrating assembly modelling, the connections between part B and part C, the support structure and the guiding plate, are specified. The designer selects compositions of the relations align, against, and fit. This renders several combinations, three of which are depicted in Figure 6.10. The parts can be assembled using two against relations and an align relation (Option A), using two alignment relations

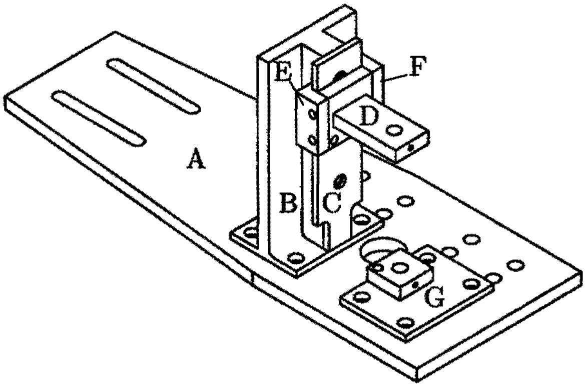


Figure 6.9: Parts positions in final assembly

and one against relation (Option B), or using three mounting assembly primitives (Option C). A mounting assembly primitive consists of a fit and an against relationship and includes the relations needed for mounting the bolt. Options A and B can be regarded variations of an abutment primitive, but no provisions are made for describing the bolts. Option C is preferred because of efficiency.

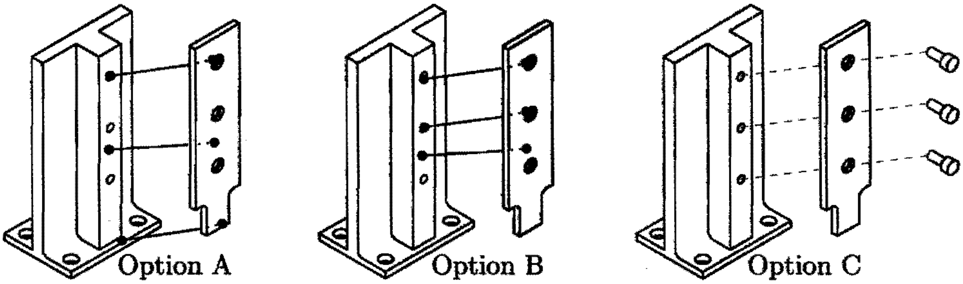


Figure 6.10: Assembling parts Band C

Relations in the primitives are created using the reference elements that were added during the modelling of the parts or reference elements that are part of the primitive, such as the elements of the bolts. Faces that resulted from previous operations but are not reference elements are not used. Table 6.2 shows the relations that define the primitive that connects parts B and C. The names of the faces are explained in Figure 6.11. Table 6.3 shows the origin of the faces.

	Reference element 1	Reference element 2	Relation
1.	face 1	face 2	against
2.	cylinder 1	cylinder 4	align
3.	cylinder 2	cylinder 5	align
4.	cylinder 3	cylinder 6	align
5.	face 4	face 3	against
6.	face 5	face 3	against
7.	face 6	face 3	against
8.	cylinder 4	cylinder 7	align
9.	cylinder 5	cylinder 8	align
10.	cylinder 6	cylinder 9	align

Table 6.2: Assembly relations

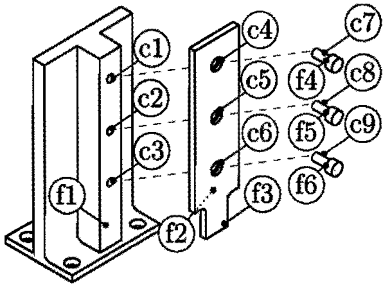


Figure 6.11: Faces used in assembly relations

no.	face	part	origin
f1	face 1	part B	bottom of box primitive used to make the front notch
f2	face 2	part C	rear of box primitive used for initial state of part C
f3	face 3	part C	front of box primitive used for initial state of part C
f4, f5, f6	face 4, 5, 6	bolt 1, 2, 3	assembly primitive, bottom sides of bolt heads
c1, c2, c3	cylinder 1, 2, 3	part B	cylinder primitives used to create holes in part B
c4, c5, c6	cylinder 4, 5, 6	part C	cylinder primitives used to create holes in part C
c7, c8, c9	cylinder 7, 8, 9	bolt 1, 2, 3	assembly primitive, shafts of bolts

Table 6.3: Origin of the faces

Besides nominal values, tolerances are added to the relations describing parts B and C. They are used to ensure that the moving holder for the upper electrode describes a straight line, without jamming. The mounting assembly primitive provides a maximum rotation as a parameter for indicating this functional constraint. After performing a worst case calculation, a rotation of 0.05° indicates a tolerance zone width of 0.15 millimetres to be allocated to flatness tolerances of faces B2 and C1. Based upon equal manufacturing effort, this value is distributed into two equal parts. Besides these flatness constraints, some other tolerances were added. They are depicted in Figure 6.12 and Table 6.4. Tolerance propagation shows the implications of the relations. The quality numbers are found by distributing the total tolerance latitude over the faces involved. The quality numbers are equal when multiple faces are involved, which means the proportion of the characteristic lengths determines the distribution among the faces. Table 6.4 shows the characteristic lengths (L), the widths of the tolerance zones (t), and the quality number derived from Figure 4.24.

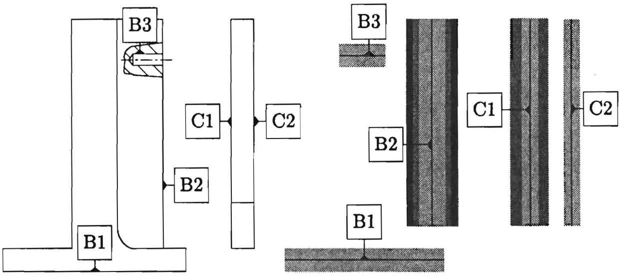


Figure 6.12: Tolerance relation in parts B and C

tolerance			L1	t1	L2	t2	B1	B2	B3	C1	C2
B1	⊥ 0.1	B2	100	0.045	146	0.055	**8.1**	**8.1**	-	-	-
B2	□ 0.075	-	146	0.75	-	-	-	8.6	-	-	-
C1	// 0.05	C2	155	0.025	155	0.025	-	-	-	**5.8**	**5.8**
C1	□ 0.075	-	155	0.075	-	-	-	-	-	8.6	-
B2	⊥ 0.1	B3	146	0.07	20	0.03	-	8.6	**8.6**	-	-

Table 6.4: Tolerances and quality numbers

From the results of the tolerance propagation, it is concluded that the flatness tolerances on faces B2 and C1 are obsolete (although they were part of the primitive). These tolerance are always met because other, stricter, tolerances must be met (perpendicularity and parallellity). Face B2 is manufactured to meet the perpendicularity constraint relative to B1 (quality 8.1), which means the tolerance zone width is 0.055 millimetres. The relation with face B3 required a width of 0.07 millimetres. By redistributing the latitude of 0.1 millimetres, the remaining 0.015 millimetres is allocated to face B3. Consequently, if face B3 has a tolerance zone width of 0.045 millimetres (thus quality 9.1) both tolerance relations are still met, but the total manufacturing costs are lower.

Design grammar

The design transformations that were described so far can be represented in the design grammar. First, the geometrical transformations for creating the initial design (sub)state, a box, and removing the rectangular pocket are addressed. After that, the assembly transformation and tolerance relations are represented.

Creating an initial design state was performed by specifying a box primitive. The relations in the primitive make sure the relative orientation of the reference element remains correct (internal relations). The geometry of the block is provided using dimensions, because no other reference elements exist to create relations with (it is the first geometrical transformation of the first design substate). The dimensions are 60×80×150 millimetres. No attributes are defined. The material of this substate is Aluminium (*Al*).

$$\begin{aligned}
\langle \mathit{Product\ Model}\rangle &\rightarrow \langle \mathit{Design\ State}\rangle \\
&\rightarrow \langle \mathit{Design\ Substate}\rangle \\
&\rightarrow \langle \mathit{Geometrical\ Transformation}\rangle\ \langle \mathit{Material}\rangle \\
&\rightarrow \langle \mathit{Geometrical\ Primitive}\rangle\ \langle \mathit{Geometrical\ Operator}\rangle\ \mathit{Al} \\
&\rightarrow \langle \mathit{Geometrical\ Primitive\ Type}\rangle\ \{\langle \mathit{Relation}\rangle\} \\
&\ [\{\langle \mathit{Dimension}\rangle\}]\ [\{\langle \mathit{Attribute}\rangle\}]\ \mathit{Add\ Al}
\end{aligned}$$

When the pocket is created, a geometrical transformation is added, so two transformations exist in the design substate. The first geometrical transformation and the material were already described.

$$\begin{aligned}
\langle \mathit{Design\ State}\rangle &\rightarrow \langle \mathit{Design\ Substate}\rangle \\
&\rightarrow \langle \mathit{Geometrical\ Transformation}\rangle \\
&\ \langle \mathit{Geometrical\ Transformation}\rangle\ \mathit{Al}
\end{aligned}$$

The second design transformation was specified using relations (implicit). The designer used a pocket primitive, a remove operator, and six relations. The only dimension left was the pocket radius.

$$\begin{aligned}
\langle \mathit{Geom.\ Transformation}\rangle &\rightarrow \langle \mathit{Geometrical\ Primitive}\rangle\ \langle \mathit{Geometrical\ Operator}\rangle \\
&\rightarrow \langle \mathit{Geometrical\ Primitive\ Type}\rangle\ \{\langle \mathit{Relation}\rangle\} \\
&\ \{\langle \mathit{Dimension}\rangle\}\ \mathit{Remove} \\
&\rightarrow \mathit{Pocket}\ \{\langle \mathit{Relation}\rangle\}\ \mathit{Radius}\ 7mm\ \mathit{Remove}
\end{aligned}$$

To illustrate the relations, two of them are described here. The first relation concerns the top of the pocket that is parallel to the left face of the box (distance 0). The second one is the pocket's virtual left plane that is parallel to the rear plane of the box (distance 0). The vector and vertex of the box faces are known and the vector and vertex of the pocket have to be calculated. The relations are described as:

$$\begin{aligned}
\langle \mathit{Relation}\rangle &\rightarrow \mathit{Plane\ False}\ (0,-1,0)\ (0,0,0)\ \langle \mathit{Envelope}\rangle\ //\ 0 \\
&\ \mathit{Plane\ False}\ \langle \mathit{Vector}\rangle\ \langle \mathit{Vertex}\rangle \\
\langle \mathit{Relation}\rangle &\rightarrow \mathit{Plane\ False}\ (-1,0,0)\ (0,0,0)\ \langle \mathit{Envelope}\rangle\ //\ 0 \\
&\ \mathit{Plane\ True}\ \langle \mathit{Vector}\rangle\ \langle \mathit{Vertex}\rangle
\end{aligned}$$

As the relation type is parallel and the distance is 0, the vector and vertex of the box faces that are referred to are copied to obtain the information for the new faces. As they are not physically present, no envelopes are defined for the pocket reference elements. The envelopes of the box reference elements depend on the default tolerances used in creating new design states, as no other tolerances were specified.

For connecting parts B and C a mounting assembly primitive was used. The number of substates is not defined at this point, as this depends on the order in which the parts and assembly relations were specified. If all parts are created first, seven design substates are present. At least the two substates representing parts B and C need to be available however.

$$\begin{aligned}
\langle \mathit{Design\ State}\rangle &\rightarrow \langle \mathit{Design\ Substate}\rangle\ \langle \mathit{Design\ Substate}\rangle\ \ldots \\
&\ \langle \mathit{Assembly\ Transformation}\rangle
\end{aligned}$$

The assembly primitive is of type mounting assembly primitive. This primitive contains 10 relations, as described in Table 6.2. The attributes are used to describe the tool-tips of the holes, threads, and the properties of the bolts.

$$\begin{aligned}
\langle \mathit{Assembly\ Transformation}\rangle &\rightarrow \langle \mathit{Assembly\ Primitive}\rangle \\
&\rightarrow \langle \mathit{Assembly\ Primitive\ Type}\rangle\ \{\langle \mathit{Relation}\rangle\} \\
&\ [\{\langle \mathit{Attribute}\rangle\}] \\
&\rightarrow \ldots
\end{aligned}$$

As an example, one of these relations is elaborated. This concerns the two faces of parts B and C that are against each other. As explained in Table 6.3, the faces used are already present. The first face concerns the bottom of the box primitive used in the design of part B, the other one concerns the rear of the box primitive used in part C. The vector and vertex describing the face are relative to the respective parts. If parts are moved to their final position in the assembly, this is described by a co-ordinate transformation that can be inferred from the assembly relationships.

$$\begin{aligned}\langle \mathit{Relation}\rangle \;\rightarrow\;& \mathit{Plane\ False}\ (0,1,0)\ (0,25,0)\ \langle \mathit{Envelope}\rangle\ \mathit{against}\\ & \mathit{Plane\ False}\ (0,1,0)\ (0,0,0)\ \langle \mathit{Envelope}\rangle\end{aligned}$$

The tolerances describe the envelopes of the faces, like the envelope of face B3 in Table 6.4. For this face, the median face is on the nominal face and the tolerance zone is 0.045 millimetres wide.

$$\begin{aligned}\langle \mathit{Envelope}\rangle &\rightarrow (\langle \mathit{Median\ Face}\rangle\ \langle \mathit{Dimension}\rangle) \mid (\langle \mathit{Boundary}\rangle\ \langle \mathit{Boundary}\rangle)\\ &\rightarrow \langle \mathit{Vector}\rangle\ \langle \mathit{Vertex}\rangle\ \langle \mathit{Dimension}\rangle\\ &\rightarrow \langle \mathit{Vector}\rangle\ \langle \mathit{Vertex}\rangle\ \langle \mathit{Dimension}\rangle\\ &\rightarrow (0,0,1)\ (40,55,130)\ 0.045\end{aligned}$$

The entire product model consists of numerous lines of design grammar like the ones mentioned above. The following sections describe how they are interpreted to create manufacturing information.

6.4 Manufacturing the assembly

The manufacturing information consists of two aspects, as mentioned in Chapters 3 and 5. First, a rough process plan used for evaluation is created. Later on, a manufacturing job has to be generated to actually create the product.

Process planning for evaluation

Process planning for evaluation of part manufacturability is illustrated using the pocket that was described in the previous sections. This part of the process planning is performed immediately after the pocket is specified. The basic information that is used to create the micro process plan is the resulting geometry, the operator, the quality of the faces, the surface roughness, and the database of machines and tools.

The procedure for creating a micro process plan depends on the primitive involved. Micro process planning is followed by reachability checking.

The pocket measures 145×95×25 millimetres and has a corner radius of 7 millimetres (which was decided by the designer). As the volume is removed, material removal operations are considered. In material removal, operations that are performed by a milling machine are considered first. For this pocket, milling is an option, as the corner radius is large enough (a tool is found with a radius smaller than 7 millimetres and a depth of cut of more than 25 millimetres) and the quality numbers are within the range of the machines available. The quality and roughness of the faces that is needed here can be achieved by a rough milling operation alone. For this operation, a pocket generally has two approach directions, one of which is available here. This is found by creating the envelope volume described by the tool and the toolholder. For one of the approach directions, this volume does not intersect with the part being manufactured. The tools volume and the collision-free direction are depicted in Figure 6.13. The volume of the pocket is used for creating NC-code later on.

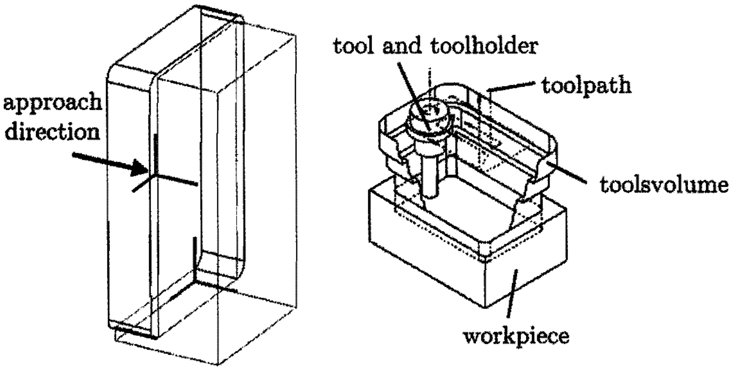


Figure 6.13: Approach direction for pocket primitive

Process planning for evaluation in assembly operations was described as part level process planning, using the operations feed, grasp, move, and mount. The connection of parts B and C is elaborated as an example.

- The feed operation is of little concern here. No non-free regions are determined, as these can be adapted to the results found for the other three operations.
- For the grasp operation, some non-free regions are identified, such as the parts of the faces that actually participate in the contact and the parts of the faces

that are close to the bolts. A number of suitable gripping faces remain. Stable grips can be established, as sets of parallel faces exist and the part is not vulnerable. The part has unsuitable magnetic properties and is too heavy for standard vacuum grippers, so a standard two finger gripper is used here. When the gripper is positioned on the available gripping faces, it appears that gripping from the side of part C is preferred. Two grips are accessible from the top, but this results in a less stable grip and a non-optimal centre of compliance (tilting part). Gripping from the front results into a grip that is less stable as well. Besides that, a collision with the bolts occurs when they are assembled.

- The move operation concerns finding a part to a location near the other part. Here, the approach direction is described as a quarter of a sphere. The part and gripper have no accessibility problems.
- The mount operation is not complex, the faces of both parts are brought into contact. Jamming or wedging does not occur, so the compliant motion is of no real concern (mechanical alignment is sufficient). An extra operation is required: mounting the bolts. Therefore, the position accuracy of the mount operation must be sufficient. In a mounting primitive, the alignment of the holes determines the assembly capability (no clearance is defined). Besides that, the gripper cannot be removed until at least one of the bolts is mounted, creating a stable subassembly.

Figure 6.14 depicts some of the grips and the approach motion set. The arrows depict the faces in the product that generated the boundaries of the approach motion set.

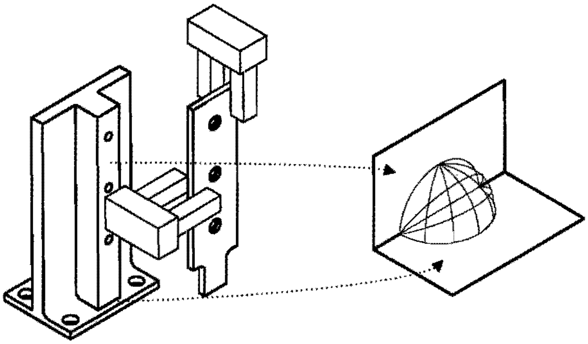


Figure 6.14: Assembly process planning of part C

Process planning for realisation

In process planning, a process plan is generated for each part separately first. Parts manufacturing process planning consists of set-up planning, fixture planning, and NC code generation. The set-up depends on the kinematics of the machine and the tolerance relations in the product. Here, a MAHO 700S 5 axis milling machine is used, of which the kinematic model is in the database.

The manufacturing operations creating faces that are in the same tolerance relation are preferably kept in the same set-up. In this case, five sides of the part can be reached in one set-up due to the presence of a B-axis in the machine. When performing process planning of this part, no tool was present that is able to reach the holes in the bottom of part B from above. Therefore, a second set-up is needed. The complex fixturing problem is overcome by letting the process planner interactively specify a fixture. Checks are implemented to avoid collision of the tools with the fixturing device and to avoid damaging vulnerable faces. For part B, a standard machine vice is sufficient. When the operations are grouped into set-ups and the machine is selected, numerical control code is generated for each part.

The part manufacturing plans are integrated in the manufacturing plan for the entire assembly. Besides these plans, information from product level assembly planning is added. Figure 6.9 depicted the positions of the parts in the final assembly. Table 6.5 shows all remaining connections that were made (+). A total of 8 assembly transformations was used to specify the entire assembly. This information is of importance when assembly sequence planning is concerned, as an Assembly State Transition is only defined when the parts actually have a connection.

	A	B	C	D	E	F	G
A	*						
B	+	*					
C	-	+	*				
D	-	-	+	*			
E	-	-	+	+	*		
F	-	-	+	+	-	*	
G	+	-	-	-	-	-	*

Table 6.5: Part connections

Figure 6.15 depicts the Assembly State Transition Diagram for the welding module. The base plate (A) is chosen as a base part, which means $\overline{ABCDEFG} \longrightarrow A/\overline{BCDEFG}$ is the first transition. Starting with part A, only part B or part G can be used for the next transition, as they are the only ones having a connection to A. This results in two possible states, $AB/\overline{CDEFG}$ and $AG/\overline{BCDEF}$. From here, new states can be created in which the part added to the assembly has a connection to at least one of the assembled parts (so state $ABC/\overline{DEFG}$ exists, while $ABD/\overline{CEFG}$ does not). This creates an ASTD that is limited in size.

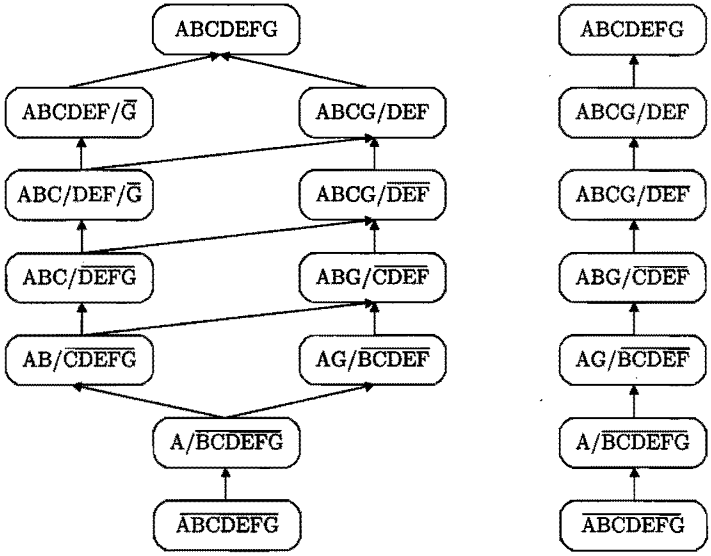


Figure 6.15: ASTD and optimal ASTD for the welding module

A state transition indicated as $ABC/\overline{DEFG} \longrightarrow ABC/DEF/\overline{G}$ describes a subassembly DEF that is created. This subassembly is created because the separate parts are not stable when assembled; no valid assembly sequence is found otherwise. As the algorithm does not detect the degree of freedom left in subassembly DEF, assembling $CDEF$ onto B is rejected for reasons of accessibility.

Optimising the ASTD according to the criteria in Chapter 5 is almost totally achieved by selecting A as the base part. The only further optimisation possible is assembling G before B according to the accessibility criterion. The optimal ASTD is also depicted in Figure 6.15.

Manufacturing grammar

For this product, the assembly sequence consists of seven assembly states (the optimal ASTD is used, as depicted in Figure 6.15).

$$\begin{aligned}\langle Manufacturing\ Job\rangle &\rightarrow \langle Assembly\ Sequence\rangle\\ &\rightarrow \langle Assembly\ State\rangle\ \langle Assembly\ State\rangle\ \langle Assembly\ State\rangle\\ &\ \langle Assembly\ State\rangle\ \langle Assembly\ State\rangle\ \langle Assembly\ State\rangle\\ &\ \langle Assembly\ State\rangle\end{aligned}$$

In the manufacturing grammar, as assembly state was defined as a collections of parts, sub-assemblies, and an assembly. The connection of parts B and C was described in so the loose parts are D, E, and F, the assembly consists of ABG and the transition moves part C from the collection of loose parts to the assembly. This results in:

$$\langle Assembly\ State\rangle \rightarrow \langle Part\rangle\ \langle Part\rangle\ \langle Part\rangle\ \langle Assembly\rangle$$

In which the assembly is:

$$\begin{aligned}\langle Assembly\rangle &\rightarrow \langle Part\rangle\ \langle Part\rangle\ \langle Part\rangle\ \langle Part\rangle\ \langle Assembly\ Task\rangle\\ &\ \langle Assembly\ Task\rangle\ \langle Assembly\ Task\rangle\ \langle Assembly\ Task\rangle\end{aligned}$$

The assembly tasks all consist of the feed operation, grasp operation, move operation, and mount operation. The move and mount operation are elaborated for the transition $ABG/\overline{CDEF} \longrightarrow ABCG/\overline{DEF}$.

$$\begin{aligned}\langle Move\ Operation\rangle &\rightarrow \langle Start\ Location\rangle\ \langle Insertion\ Point\rangle\ \langle Trajectory\rangle\\ \langle Mount\ Operation\rangle &\rightarrow \langle Insertion\ Point\rangle\ \langle Insertion\ Path\rangle\end{aligned}$$

In the design grammar, the assembly information was presented as six alignment relations and four against relations. The reference elements in these assembly relations are described in the parts ($\upuparrows$=align, $\veebar$=against).

$$\begin{aligned}\langle Assembly\ Transformation\rangle &\rightarrow Mounting\\ &\ \langle Reference\ Element\rangle\ \upuparrows\ \langle Reference\ Element\rangle\\ &\ \ldots\\ &\ \langle Reference\ Element\rangle\ \veebar\ \langle Reference\ Element\rangle\end{aligned}$$

From the geometrical informations in the part and the assembly relations, it can be concluded that the insertion point is (40, 65,75). The trajectory for the move operation is the quarter sphere, depicted in Figure 6.14. The insertion path is a straight line with direction (0,1,0), using mechanical alignment.

6.5 Evaluation

Although the subject of the case was not entirely realistic, the design session shows that a product can be designed and manufactured using a design support tool that is based upon the concepts resulting from this research. The relations can be described in the tool. A designer does not have to find out what dimensions are needed to create a volume that renders the desired result. Other aspects that formerly created problems, such as the influence of the sequence of the rotations about the three axes, the location of the axis-system on a volume, and the possibility to make volumes appear upside down or inside out, are no longer relevant. Besides that, choosing from a large number of primitives that are almost equal (primitives that are in fact instances) is no longer needed.

The algorithms that determine whether a valid volume is defined work very well. However, feedback on the reference elements specified and the volumes created is critical and the method strongly depends on well-defined primitives. The tool forces the designer to find out what dimensions are essential to functionality. The tolerance information in these relations is added as a matter of course, that is, when designers are not forced to adapt to the notations used in drawings the deviant rules for tolerance analysis go unnoticed. The fact that tolerances are not directed and regarded as constraints does not meet any resistance. Still, the absence of quantitative support for creating tolerance relations devaluates the merit of the current tool.

The time needed for designing a product is relatively short. The case presented indicates that the time needed to design a ready-to-manufacture product, that is, including process planning, approaches the time to create conventional drawings of a single part. This is true provided the parts are pseudo-prismatic parts (like the parts in the case presented). Furthermore, either the designer is an experienced user of the tool, or the tool is made more user friendly. At present, an educated guess is made to obtain the time needed for definition of assembly relations, as no implementation is provided.

Considering the assembly planning performed in this case, which relies on imaginary aspects of the tool, one may consent to optimism. Selecting a base part renders an enormous reduction of complexity (theoretically, 7!=5040 combinations are possible, but only 11 nodes are generated). Even if no wide array of assembly primitives is worked out, a product can be provided with assembly relations.

Chapter 7

Conclusions and recommendations

The observation that designers need to provide product designs in short design cycles and face an increasing amount of information initially motivated this research. The objective of the research was to improve the creation of assembled products by taking manufacturing aspects into consideration during design to reduce costs and lead times and to increase quality.

This research aimed at integrating part design and assembly operations. Besides the part properties, assembly information and tolerance information had to be incorporated in the resulting product models. The concepts that were developed had to eventually lead to the creation of a design support tool.

7.1 Conclusions

By selecting a range of products and focusing on the relation between a product and its production system, a target for design support was established: creating product models that are free of manufacturability errors and automating the generation of manufacturing information.

A model of the product creation process, which represents the activities that transform requirements into products, is suitable for explaining the approach that meets the target: manufacturability evaluation during design. If every data element or decision added by a designer is evaluated for manufacturability, a product model results that can be manufactured right first time. This evaluation can be automated, but is limited to geometrical design and a certain type of products. The creative part of design cannot be formally described or automated, and the model does not elaborate on finding errors other than manufacturability errors. The approach results in a less strict separation between design and process planning activities.

When providing more details on the design process, this research showed that the concept of states and state transitions is suitable for describing the geometrical design process. A state transition is evaluated to obtain information about its manufacturing. A design history composed of the states and state transitions serves as a product model.

Evaluating manufacturability requires process planning. Automated process planning in turn requires the selection of manufacturing processes and tools. Algorithms were provided that compare manufacturing complexity, described by characteristic dimensions of the product and the tolerance latitude, and factory capability (described in the same terms and extended with information on available equipment).

For actually specifying the details of a product, geometric modelling concepts were provided through this research. An alternative to conventional product models was obtained by using faces as the primitives of design. These faces can be related to each other, thus creating the possibility to model parts, assemblies, and tolerances. This approach diminishes some of the problems currently encountered in tolerance representation and assembly. It also allows modelling in terms of critical dimensions and functional relations. Part geometry, assembly operations, and tolerance information are integrated into a single product model.

The research provided a more formal approach to designing and manufacturing assemblies by means of a design grammar and a manufacturing grammar. The grammars indicate that the symbols used are unambiguous and coherent. They also serve as a specification for implementing a design support tool that evaluates manufacturability of a design. As the main problem in assembly proved to be the lack of knowledge on the processes (that is, knowledge in the correct form for assessing manufacturability), a consistent set of assembly process planning symbols was provided, including a significant part of the knowledge needed for manufacturability evaluation.

A satisfying description of the assembly process was found in the use four standard assembly operations: feed, grasp, move, and mount. In this research, the validation of manufacturability was extended from parts manufacturing to assembly by separately checking these assembly operations for validity. Some provisions are needed to reduce complexity. These provisions concern volume, weight, assembly directions, the number of sub-assemblies, and reversibility. A representation for the class of products that remains was found in the Assembly State Transition Diagram. Finding collision free paths in assembly proves to be the most complex operation, but solutions were found for this problem.

The research resulted in a proposal for a design support system that interprets a product model and creates the corresponding manufacturing information. The fact that evaluation of manufacturability during design is possible, indicates how the functionality of a design support tool is increased compared to conventional tools. An implementation of this design support tool indicated that it is suited for designing actual products. The case study illustrated the use of relations to model part geometry, assembly, and tolerances. It also demonstrated how a product model is interpreted to create manufacturing information.

The research shows that the definitions of design primitives have far-reaching consequences. Design and manufacturability assessment of products strongly depends on well-defined primitives. It is also clear that the designers must be provided with sufficient feedback on their actions to avoid that they loose track of what they are doing.

7.2 Recommendations

The envisioned benefits concerning improvement of lead time are indicated in this research. These benefits need to be confirmed by extensive empirical studies. Although the actual design of design support software was not the main objective of the research, a complete implementation of the design support system is needed for performing such studies. The mappings to manufacturing process are still unsuited for more than proofing the concepts. If a fully functional design support tool is available, a fair comparison of the concepts presented here and conventional techniques is possible. Comparing the results of the design of a number of relevant products is needed to prove the envisioned merits of the new approach.

The design grammar presented attempts to formalise the description of an artifact. This approach can be pursued, along with further attempts to formalise and document knowledge on manufacturing processes. If the design grammar is used as a basis for the implementation, it benefits from an attempt to build an implementation using it in its current form.

Implementation of details and optimisations like in generating numerical control code is of less concern, as sufficient tools are available. The implementation then serves as a front-end to these tools, that is, they are fed with design representations that are verified. During conversion however, information will be lost, so ultimately, the details should also be provided. Completely discarding the details at first is not wise however, because the tool is then unable to encompass an entire product creation process.

Regarding tolerance propagation, which is a crucial element in this research, it is worth researching if equal manufacturing effort truly delivers the lowest manufacturing costs in all cases. So far, it was made plausible, but comparing different propagation algorithms for a larger series of product is useful. Besides that, a lot of work is to be done in finding qualitative and quantitative support for the definition of tolerances. Tolerance relations based on the designers' experience or trial and error do not render an optimal solution concerning the compromise between the functions a product has to perform and its manufacturing costs.

Selection of assembly processes and equipment was not sufficiently covered. A series of projects comparable to the one that provided the capability numbers for material removal should be carried out to generate numerical data on performance of assembly processes. Besides that, attention should be paid to the types of operations that were not elaborated here, such as for non-reversible connections or snap connections.

It was stated that errors that originate from sources that are not expressed in the design requirements or the geometrical design cannot be found. Therefore, an interface between the design support tool and other tools, such as predictive modelling tools, should be established. This expands the applicability of the tool and thus promotes its use in early stages of the product creations process. If the design support tool for manufacturability evaluation is presented as a stand-alone tool, the risk of it being treated as an aberrant and thus time consuming tool is increased.

Bibliography

[Aasl93] K. Aasland, Design history in connection with the domain theory, *Proceedings of the ICED '93*, Volume 3, 1993, (pp. 1385-1392).

[Ambl75] A.P. Ambler, R.J. Popplestone, Inferring the positions of bodies from specified spatial relationships, *Artificial Intelligence* 6, 1975, (pp. 157-174).

[Andr88] M.M. Andreasen, S. Kähler, T. Lund, K.G. Swift, *Design for Assembly*, IFS publications/Springer-Verlag, 1988.

[Arba89] F. Arbab, Design object modeling, *Intelligent CAD I*, Proceedings of the IFIP TC 5/WG 5.2 Workshop, 1987, North-Holland, 1989, (pp. 5-12).

[Arba90] F. Arbab, B. Wang, Reasoning about geometric constraints, *Intelligent CAD II*, Proceedings of the IFIP TC 5/WG 5.2 Workshop, September 1988, North-Holland, 1990, (pp. 93-107).

[Aren89] J.H.A. Arentsen, *Factory Control Architecture, A Systems Approach*, Ph.D. Thesis, Eindhoven University of Technology, The Netherlands, 1989.

[Aren96] N.W.A. Arends, *A Systems Engineering Specification Formalism*, Ph.D. Thesis, Eindhoven University of Technology, The Netherlands, 1996.

[Baar95] J.P. Baartman, *Automation of Assembly Operations on Parts*, Ph.D. Thesis, Delft University of Technology, The Netherlands, 1995.

[Bake92] R. Bakerjian, P. Mitchell, *Design for Manufacturability*, Tool and manufacturing engineers handbook, Volume VI, SME, USA, 1992.

[Baxt92] J.E. Baxter, N.P. Juster, A. de Pennington, An assessment of assembly mating conditions in the context of a product model, *Computers in Engineering*, Volume 1, 1992, (pp. 421-428).

[Bert95] G.R. Bertoline, E.N. Wiebe, C.L. Miller, L.O. Nasman, *Engineering Graphics Communication*, Irwin Graphics Series, 1995.

[Bhot91] K.R. Bhote, *World Class Quality; Using Design of Experiments to Make it Happen*, American management association, New York, USA, 1991.

[Biro94] A. Birolini, *Quality and Reliability of Technical Systems*, Springer Verlag, Berlin, Germany, 1994.

[Bjør78] Ø. Bjørke, *Computer-Aided Tolerancing*, Tapir publishers, Trondheim, Norway, 1978.

[Bles94] L.T.M. Blessing, *A Process-Based Approach to Computer-Supported Engineering Design*, Ph.D. Thesis, University of Twente, The Netherlands, 1994.

[Boer90] J.R. Boerma, *The Design of Fixtures for Prismatic Parts Manufacturing*, Ph.D. Thesis, University of Twente, The Netherlands, 1990.

[Bolw91] P.T. Bolwijn, T. Kumpe, *Marktgericht Ondernemen*, Stichting Management studies, 's-Gravenhage, 1991. (In Dutch.)

[Book63] P.J. Booker, *A History of Engineering Drawing*, Chatto & Windus, London, Great Britain, 1963.

[Boot82] G. Boothroyd, *Design for Assembly Handbook*, University of Massachusetts, 1982.

[Bosh92] D.C. Boshuisen, *Het Inkomen van de Onderneming*, Inauguration speech, Eindhoven University of Technology, 1992. (In Dutch.)

[Brai74] I.C. Braid, *Designing with Volumes*, Cantab Press, Cambridge, UK, 1974.

[Bran93] L.E.M.W. Brandts, *Design of Industrial Systems*, Ph.D. Thesis, Eindhoven University of Technology, The Netherlands, 1993.

[Brez96] J.C. Brezet, *Op Weg naar Duurzame Productontwikkeling*, Inauguration speech, Delft University of Technology, 1996. (In Dutch.)

[Bron93] W.F. Bronsvoort, F.W. Janssen, Feature modelling and conversion: Key concepts to concurrent engineering, *Computers in Industry*, 21(1), 1993, (pp. 61-86).

[Brow92] K.N. Brown, J.H. Sims Williams, Grammars of features in design, *Artificial Intelligence in Design '92*, Kluwer, 1992, (pp. 287-306).

[Brow95] K.N. Brown, C.A. McMahon, J.H. Sims Williams, Features aka the semantics of a formal language of manufacturing, *Research in Engineering Design*, Volume 7, Number 3, 1995, (pp. 151-172).

[Brow96] J. Brownie, J. Harhen, J. Shivnan, *Production Management Systems, an Integrated Perspective*, Addison-Wesley Publishing, Great Britain, 1996.

[Byr87] D.M. Byrne, S. Taguchi, The Taguchi approach to parameter design, *Quality Progress*, December 1987, (pp. 19-26).

[Caga92] J. Cagan, T.R. Kurfess, Optimal tolerance allocation over multiple manufacturing alternatives, *Advances in Design Automation*, Volume 44-2, ASME, 1992, (pp. 165-172).

[Cham93] M.A. Chamberlain, A. Joneja, T.C. Chang, Protrusion-features handling in design and manufacturing planning, *Computer Aided Design*, Volume 25, Number 1, 1993, (pp. 19-28)

[Chen90] A. Chen, B. McGinnis, D.G. Ullman, T.G. Dietterich, Design history knowledge representation and its basic computer implementation, *Design Theory and Methodology-DTM'90*, ASME, 1990, (pp. 175-184).

[Coum95] L.H.J. Coumans, *Templates voor Implicit Locating*, Report WPA 210036, Eindhoven University of Technology, The Netherlands, 1995. (In Dutch.)

[Crev96] C.M. Creveling, *Tolerance Design, a Handbook for Developing Optimal Specifications*, Addison Wesley Longman, Reading, MA, USA, 1996.

[Cros89] N. Cross, *Engineering design methods*, The Open University, Milton Keynes, UK, Wiley & Sons, 1989.

[Cutk85] M. Cutkosky, *Robot Grasping and Fine Manipulation*, Kluwer, USA, 1985.

[Cutk93] M. Cutkosky, R. Engelmore, R. Fikes, M. Genesereth, T. Gruber, PACT, an experiment in integrating concurrent engineering systems, *IEEE Computer*, Volume 26, Number 1, 1993, (pp. 28-37).

[DeFa93] T.L. De Fazio, A.C. Edsall, R.E. Gustavson, J. Hernandez, et. al., A prototype of feature-based design for assembly, *Journal of Mechanical Design*, Volume 115, ASME, December 1993, (pp. 724-734).

[Delb89] F.L.M. Delbressine, *On the Integration of Design and Manufacturing*, Ph.D. Thesis, Eindhoven University of Technology, The Netherlands, 1989.

[Delc92] A. Delchambre, *Computer Aided Assembly Planning*, Chapman & Hall, London, UK, 1992.

[Desr95] A. Desrochers, R. Maranzana, Constrained dimensioning and tolerancing assistance for mechanisms, *CIRP Seminar on Computer Aided Tolerancing*, 1995, (pp. 21-29).

[ElMa93] H.A. ElMaraghy, Evolution and future perspectives of CAPP, *Annals of the CIRP*, Volume 42/2, 1993, (pp. 739-751).

[Erve88] A.H. van 't Erve, *Generative Computer Aided Process Planning for Part Manufacturing, an Expert System Approach*, Ph.D. Thesis, University of Twente, The Netherlands, 1988.

[Feld96] K. Feldmann, H. Rottbauer, N. Roth, Relevance of assembly in global manufacturing, *Annals of the CIRP*, Volume 45/2, 1996.

[Ferr90] P.M. Ferreira, S.C. Lu, X. Zhu, *A Conceptual Model for Process Planning*, CAM-I report, University of Illinois, USA, 1990.

[Fing89a] S. Finger, J.R. Dixon, A review of research in mechanical engineering design I: Descriptive, prescriptive, and computer based models of design processes, *Research in Engineering Design*, Volume 1, 1989, (pp. 51-67).

[Fing89b] S. Finger, J.R. Dixon, A review of research in mechanical engineering design II: Representations, analysis, and design for the life cycle, *Research in Engineering Design*, Volume 1, 1989, (pp. 121-137).

[Ford93] Ford Motor Company, *Design FMEA*, FMEA manual and design FMEA form No. 1695, 1993.

[From92] B. Fromont, D. Sriram, Constraint satisfaction as a planning process, *Artificial Intelligence in Design '92*, Kluwer, 1992, (pp. 97-117).

[Gada94] M.H. Gadallah, H.A. ElMaraghy, The tolerance optimization problem using a system of experimental design, *Advances in Design Automation*, Volume 69-1, ASME, 1994, (pp. 251-265).

[Groo93a] R. de Groot, *Het Automatisch Genereren van Set-ups en Fixtures binnen IDM process planning*, Report WPA1553, Eindhoven University of Technology, The Netherlands, 1993. (In Dutch.)

[Groo93b] R. de Groot, F.L.M. Delbressine, Knowledge based design is duur maar de moeite waard, *Mechanische Technologie*, Mei 1993, (pp. 32-36). (In Dutch.)

[Grub92] T.R. Gruber, *Ontolingua: A Mechanism to Support Portable Ontologies*, Stanford University, Knowledge Systems Laboratory, Technical Report KSL-91-66, March 1992.

[Hash93] F.M. Hashim, N.P. Juster, A. de Pennington, Generating design variants based on functional reasoning, *Proceedings of the ICED '93*, Volume 1, 1993, (pp. 60-66).

[Heem90] C.J.M. Heemskerk, *A Concept for Computer Aided Process Planning for Flexible Assembly*, Ph.D. Thesis, Delft University of Technology, The Netherlands, 1990.

[Held91] M. Held, *On the Computational Geometry of Pocket Machining*, Springer Verlag, Salzburg, 1991.

[Henz95] G. Henzold, *Handbook of Geometrical Tolerancing, Design, Manufacturing and Inspection*, John Wiley & Sons, Chichester, England, 1995.

[Hito96] K. Hitomi, *Manufacturing Systems; a Unified Approach to Manufacturing Technology, Production Management, and Industrial Economics*, Taylor & Francis, London, Great Britain, 1996.

[Holl95] M. Holland, Prozeßgerechte Toleranzfestlegung, Bereitstellung von Prozeßgenauigkeitsinformationen für die Konstruktion, *VDI Fortschrittberichte*, Reihe 20, Nummer 137, VDI Verlag, 1995. (In German.)

[Home91] L.S. Homem de Mello, A.C. Sanderson, Representation of Mechanical Assembly Sequences, *IEEE Transactions on Robotics and Automation*, Volume 7, Number 2, 1991, (pp. 211-227).

[Hout91] F.J.A.M. van Houten, *PART: A Computer Aided Process Planning System*, Ph.D. Thesis, University of Twente, The Netherlands, 1991.

[Hua92] K. Hua, I. Smith, B. Faltings, Adaptation of spatial design cases, *Artificial Intelligence in Design '92*, Kluwer, 1992, (pp. 559-575).

[Hubk73] V. Hubka, *Theorie der Maschinensysteme*, Springer Verlag, Berlin, Germany, 1973.

[Hubk76] V. Hubka, *Theorie der Konstruktionsprozesse*, Springer Verlag, Berlin, Germany, 1976.

[Huus95] P. Huuskonen, K. Kaarela, Explaining plant design knowledge through means-end modelling, *6th International Conference on Human-Computer Interaction*, July 1995.

[Jaco96] P.F. Jacobs, *Stereolithography and Other RP&M Technologies*, ASME Press, 1996.

[Kawa93] Y. Kawashima, S. Nonaka, K. Itoh, S. Tokomasu, Product framework model with structured expression of surface constraints, *Proceedings of ICED '93*, 1993, (pp. 1533-1540).

[Kess89] K. Kessel-Hunter, Knowledge-based engineering case study: Jet engine turbine blade design, *SME Autofact '89 conference*, 1989, (pp. 26.1-26.19).

[Kimu92] F. Kimura, H. Suzuki, K. Takahashi, Product design evaluation based on effect of shape errors for part assembly, *Annals of the CIRP*, Volume 41/1, 1992, (pp. 193-196).

[Ko87] H. Ko, K. Lee, Automatic assembling procedure generation from mating conditions, *Computer-Aided Design*, Volume 19, No.1, 1987, (pp. 3-10).

[Kotl88] P. Kotler, *Marketing Management, Analysis, Planning, Implementation, and Control*, Prentice-Hall, New Jersey, USA, 1988.

[Krik92] J.W.M. Krikhaar, *Design for Mass Production of Small Lotsize Mechanical Systems*, Ph.D. Thesis, Eindhoven University of Technology, The Netherlands, 1992.

[Kris94] M. Krishnaswami, R.W. Mayne, Optimizing tolerance allocation based on manufacturing quality loss, *Advances in Design Automation*, Volume 69-1, ASME, 1994, (pp. 211-217).

[Krom93] R. Krom, *Een Tolerantie-Modeller in een CAD/CAM Omgeving*, Report WPA1608, Eindhoven University of Technology, The Netherlands, 1993. (In Dutch.)

[Kutt93] D. Kuttig, Potential and limits of functional modelling in the CAD process, *Research in Engineering Design*, Volume 5, 1993, (pp. 40-48).

[Lato91] J.C. Latombe, *Robot Motion Planning*, Kluwer, Boston, USA, 1991.

[Lee85a] K. Lee, D.C. Gossard, A hierarchical data structure for representing assemblies, *Computer-Aided Design*, Volume 17, Number 1, 1985, (pp. 15-19).

[Lee85b] K. Lee, G. Andrews, Inference of the positions of components in an assembly, *Computer-Aided Design*, Volume 17, Number 1, 1985, (pp. 20-23).

[Liba88] E.C. Libardi, J.R. Dixon, M.K. Simmons, Computer environments for the design of mechanical assemblies: A research review, *Engineering with Computers 3*, 1988, (pp. 121-136).

[Liu91] H.C. Liu, B.O. Nnaji, Design with spatial relationships, *Journal of Manufacturing Systems*, Volume 10, Number 6, 1991, (pp. 449-463).

[Loud93] K.C. Louden, *Programming Languages: Principles and Practice*, PWS-Kent publishing, Boston, MA, USA, 1993.

[Mänt91] M. Mäntylä, WAYT: Towards a modeling environment for assembled products, *Intelligent CAD III*, Proceedings of the IFIP TC 5/WG 5.2 Third Workshop, September 1989, North-Holland, 1991, (pp. 187-204).

[Mart91] P. Martens, *CAM/CAM for Assembly Planning*, Ph.D. Thesis, Delft University of Technology, The Netherlands, 1991.

[Merc61] M.E. Merchant, The manufacturing system concept in production engineering research, *Annals of the CIRP*, Volume 10/2, 1961 (pp. 77-83).

[Net94] A.J. van der Net, *Implicit Locating for Specifying Geometrical Objects in IDM*, Stan Ackermans Institute, Eindhoven University of Technology, The Netherlands, 1994.

[Net95] A.J. van der Net, F.L.M. Delbressine, *Implementation of the Implicit Locating Algorithms in IDM*, Internal notes, Eindhoven University of Technology, The Netherlands, 1995.

[Net96] A.J. van der Net, W.A.H. de Vries, F.L.M. Delbressine, A.C.H. van der Wolf, A relation based product model for integrating design and manufacturing, *Annals of the CIRP*, Volume 45/1, 1996 (pp. 161-164).

[Nevi89] J.L. Nevins, D.E. Whitney, T.L. De Fazio, A.C. Edsall, R.E. Gustavson, R.W. Metzinger, W.A. Dvorak, *Concurrent Design of Products and Processes*, McGraw-Hill, USA, 1989.

[Nnaj90] B.O. Nnaji, H.C. Liu, Feature reasoning for automatic robotic assembly and manufacturing in polyhedral representation, *International Journal for Production Research*, Volume 28, Number 3, 1990, (pp. 517-540).

[Onos89] M. Onosato, H. Yoshikawa, A framework on formalization of design object for intelligent CAD, *Intelligent CAD I*, Proceedings of the IFIP TC 5/WG 5.2 Workshop, 1987, North-Holland, 1989, (pp. 5-12).

[Phil94] Philips Machinefabriek-M, en nu de praktijk..., *Themadag Doelmatiger Monteren*, Lecture Notes, Mikrocentrum Nederland, Eindhoven, The Netherlands, 1994. (In Dutch.)

[Port80] M.E. Porter, *Competitive Strategy, Techniques for Analyzing Industries and Competitors*, The Free Press, New York , USA, 1980.

[Rank97] A.M. Rankers, *Machine Dynamics in Mechatronic Systems*, Ph.D. Thesis, University of Twente, The Netherlands, 1997.

[Requ83] A.A.G. Requicha, Towards a theory of geometric tolerancing, *International Journal of Robotics Research*, Volume 2, Number 4, 1983, (pp. 45-60).

[Requ84] A.A.G. Requicha, Representation of tolerances in solid modelling: Issues and alternative approaches, *Solid Modelling by Computers: From Theory to Applications*, Plenum Press, New York, USA, 1984.

[Requ96] A.A.G. Requicha, Geometric reasoning for intelligent manufacturing, *Communications of the ACM*, Volume 39, Number 2, 1996, (pp. 71-76).

[Roch87] D.N. Rocheleau, K. Lee, System for interactive assembly modelling, *Computer-Aided Design*, Volume 19, Number 2, 1987, (pp. 65-72).

[Rood84] J.E. Rooda, J.H.A. Arentsen, Ein Strukturmodell zur Beschreibung von Transport- und Produktionssystemen, *f+h - Fördern und Heben*, Volume 34, Number 9, 1984, (pp. 649-651). (In German.)

[Rood96] J.E. Rooda, *The Modelling of Industrial Systems*, Lecture Notes, Eindhoven University of Technology, The Netherlands, 1996.

[Rooz95] N.F.M. Roozenburg, J. Eekels, *Product Design: Fundamentals and Methods*, John Wiley & Sons, Chicester, England, 1995.

[Salo95] O.W. Salomons, H.J. Jonge Poerink, F. van Slooten, F.J.A.M. van Houten, H.J.J. Kals, A computer aided tolerancing tool based on kinematic analogies, *CIRP Seminar on Computer Aided Tolerancing*, 1995, (pp. 53-72).

[Salu91] F.A. Salustri, R.D. Venter, Towards a logical theory of engineering design information, *Computers in Engineering*, Volume 1, 1991, (pp. 161-167).

[Sars95] G.W. Sars, *Integratie Kwaliteitscontrole in IDM*, Report WPA 310018, Eindhoven University of Technology, The Netherlands, 1995. (In Dutch.)

[Sche98] P.H.J. Schellekens, P.C.J.N. Rosielle, et. al., *Design for Precision: Current Status and Trends*, Keynote paper CIRP, 1998. (To be published.)

[Scho95] R.M.M. Schoufs, *CAD-CAM/CMM: Haalbaarheid en Implementatie*, Report WPA 210028, Eindhoven University of Technology, The Netherlands, 1995. (In Dutch.)

[Shah88] J.J. Shah, M.T. Rogers, Expert form feature modelling shell, *Computer Aided Design*, Volume 20, Number 9, 1988, (pp. 515-524).

[Shah93] J.J. Shah, M.T. Rogers, Assembly modelling as an extension of feature-based design, *Research in Engineering Design*, Volume 5, Numbers 3&4, 1993, (pp. 218-237).

[Shah95] J.J. Shah, M. Mäntylä, *Parametric and Feature-Based CAD/CAM*, John Wiley & Sons, 1995.

[Shal92] D. Shalon, D. Gossard, K. Ulrich, D. Fitzpatrick, Representing geometric variations in complex structural assemblies on CAD systems, *Advances in Design Automation*, Volume 44-2, ASME, 1992, (pp. 121-132).

[Sheu93] L.C. Sheu, J.T. Lin, Representation scheme for defining and operating form features, *Computer Aided Design*, Volume 25, Number 6, 1993, (pp. 333-347).

[Smit92] G.H. Smit, *A Hierarchical Control Architecture for Job-shop Manufacturing systems*, Ph.D. Thesis, Eindhoven University of Technology, The Netherlands, 1992.

[Sodh91] R. Sodhi, J.U. Turner, Representing tolerance and assembly information in a feature-based design environment, *Advances in Design Automation*, Volume 32-1, ASME, 1991, (pp. 101-108).

[Suh90] N.P. Suh, *The Principles of Design*, Oxford University Press, USA, 1990.

[Sull86] L.P. Sullivan, Quality function deployment, *Quality Progress*, June 1986, (pp. 39-50).

[Swif97] K.G. Swift, J.D. Booker, Design capability and the cost of failure, *Journal of Engineering Manufacture*, Volume 211, Number B6, 1997, (pp. 409-423).

[Taka93] K. Takahashi, H. Suzuki, F. Kimura, Motion analysis of parts with geometrical errors based on dynamic simulation, *Proceedings of 3rd CIRP Seminars on Computer Aided Tolerancing*, Eyrolles, 1993, (pp. 85-95).

[Tomi89] T. Tomiyama, T. Kiriyama, H. Takeda, D. Xue, H. Yoshikawa, Metamodel: A key to intelligent CAD systems, *Research in Engineering Design*, Volume 1, Number 1, 1989, (pp. 19-34).

[Ullm89] D.G. Ullman, A taxonomy of mechanical design, *Design Theory and Methodology - DTM'89*, ASME, 1989, (pp.23-36).

[Ullm92] D.G. Ullman, *The Mechanical Design Process*, McGraw-Hill, 1992.

[Ullm93] D.G. Ullman, A new view on function modeling, *Proceedings of the ICED '93*, Volume 1, 1993, (pp. 21-28).

[Ulri95] K.T. Ulrich, S.D. Eppinger, *Product Design and Development*, McGraw-Hill, New York, USA, 1995.

[VDI87] VDI, *Guideline-2221: Systematic Approach to the Design of Technical Systems and Products*, VDI Verlag, Düsseldorf, Germany, 1987.

[Veer89] P.J. Veerkamp, Multiple worlds in an intelligent CAD system, *Intelligent CAD I*, Proceedings of the IFIP TC 5/WG 5.2 Workshop, October 1987, North-Holland, 1989, (pp. 77-88).

[Veld72] J. in 't Veld, Denken in systemen: Systeembegrippen, systeembenadering, hiërarchie van systemen, *De Ingenieur*, Volume 84, Number 32/33, 1972, (pp. 680-685). (In Dutch.)

[Voel77] H.B. Voelcker, A.A.G. Requicha, Geometric modeling of physical parts and processes, *IEEE Computer*, Volume 10, Number 12, 1977, (pp. 48-57).

[Vrie92] W.A.H. de Vries, *Ontwerpregels voor CAD/CAM*, Report WPA 1285, Eindhoven University of Technology, The Netherlands, 1992. (In Dutch.)

[Vrie96] W.A.H. de Vries, *Manufacturable Design Operations*, Ph.D. Thesis, Eindhoven University of Technology, The Netherlands, 1996.

[Will97] M.A. Willemse, *Interactive Product Design Tools for Automated Assembly*, Ph.D. Thesis, Delft University of Technology, The Netherlands, 1997.

[Whit96] D.E. Whitney, Why mechanical design cannot be like VLSI design, *Research in Engineering Design*, Volume 8, Number 3, 1996, (pp. 125-138).

[Wolt88] J.D. Wolter, *On the Automatic Generation of Plans for Mechanical Assembly*, Ph.D. Thesis, University of Michigan, USA, 1988.

[Wolt91] J. Wolter, P. Chandrasekaran, A concept for a constraint-based representation of functional and geometric design knowledge, *Solid Modelling Foundations and CAD/CAM Applications*, ACM, 1991, (pp. 409-418).

[Woo90] T.C. Woo, B.F. von Turkovich, Visibility Map and its Application to Numerical Control, *Annals of the CIRP*, Volume 39/1, 1990, (pp. 451-454).

[Wood89] R.F. Woodbury, I.J. Oppenheim, An approach to geometric reasoning, *Intelligent CAD I*, Proceedings of the IFIP TC 5/WG 5.2 Workshop, October 1987, North-Holland, 1989, (pp. 149-168).

[Xue92] D. Xue, H. Takeda, T. Kiriyama, T. Tomiyama, H. Yoshikawa, An intelligent integrated interactive CAD, *Intelligent Computer Aided Design*, IFIP, North-Holland, 1992, (pp. 163-192).

[Yosh93] H. Yoshikawa, Systemization of design knowledge, *Annals of the CIRP*, Volume 42/1, 1993, (pp. 131-134).

Index

Appendix A

Rapid prototyping

Rapid prototyping concerns fast generation of objects directly from a CAD database. Because its application changed from mere visualisation to actual manufacturing, the terminology has changed from rapid prototyping to rapid prototyping and manufacturing. The processes used are layer-additive processes. At the moment, about 1000 rapid prototyping and manufacturing systems operate world-wide [Jaco96].

A.1 Rapid prototyping processes

Rapid prototyping processes use a (de facto) standard file format: the STereoLithography file or STL-file, after the first commercially available process (1987). It represents the surface of an object as a series of triangles. The STL-file is sliced to acquire cross-sections of the object. The result is a SLIce-file (SLI-file) that is used by the processes. A problem in rapid prototyping is the limited accuracy due to shrinkage of the materials and stair-stepping errors caused by finite layer thickness, although recent developments have increased stereolithography accuracy. For most other processes, no reliable accuracy data is available.

Stereolithography

Stereolithography (SL or SLA) uses photocurable resin. Parts are built by directing an ultraviolet laser over the surface of a vat of liquid resin, curing slices of the part. First, the boundaries are traced, followed by hatching to make the cross section solid. Between slices, the part is lowered into the vat by the slice distance. When the part is finished, support structures are removed if needed and the excess resin is cleaned off. Finally, the object is flooded with ultraviolet light for full strength. Dual beam laser curing is a similar process, in which photopolymer cures at the point of intersection of two laser beams, different in wavelength. There is also an SL Quick Cast technology; cross-hatching of SL parts to allow them to be used in casting processes (like the lost wax process) or short run tooling.

Laminated object manufacturing

The LOM process uses (low-cost) paper or plastic slices, covered with pressure and temperature sensitive adhesives. They are laminated together and cross-sections are cut out with an infrared laser. The adhesive is activated using a heated roller. The excess material is immediately removed or left as a support. If it is not removed, it is diced by the laser and broken away later. Parts come out with a wood-like texture and can be used as injection tools, concept models, or casting models. In a similar process, Computer Aided Manufacturing of Laminated Engineering Materials (CAM-LEM), cross-sections are cut from sheets, robotically assembled, and laminated by a suitable method.

Selective laser sintering

Selective Laser Sintering (SLS) bonds powders with a high power laser. A part is built inside of a cylinder that contains a moveable platform with a layer of powder. The heat from the laser causes the particles to soften and bond together, creating a layer of the part. Between layers, the platform drops by the height of one layer. Various materials are used, like waxes, nylons, or polycarbonate. The unused powder serves as a support structure and can be reused after removal. Instead of a two step process (deposition and bonding) a one step process can be used, where the metal powder is injected into the operation spot and liquefied by a laser beam.

Solid ground curing

In Solid Ground Curing (SGC), the cross section information is used to create a photo-mask (similar to xerography). The part that represents solid material remains transparent. A layer of photopolymer is spread on a work surface and ultra-violet light is projected through the photo-mask. The exposed resin hardens and the unaffected resin is removed. Next, the surface is coated with liquid wax, filling the cavities from the unaffected resin. A chilling plate hardens the wax and the combined structure is milled to the correct thickness, providing a work surface for the next layer. The wax is removed by melting or rinsing.

Fused deposition modelling

In Fused Deposition Modelling (FDM), a temperature-controlled head extrudes layers of thermoplastic material. Material filament is fed into the head, which deposits the heated material into place with precision. Various thermoplastic materials and waxes are used. The Advanced Material Fused Deposition Modelling system (AMFDM) uses a high-pressure head and material supplied in feed stock form. The advantage is the materials capability; it can build with materials like Alumina, Zirconia, PMMA, and various other plastics and ceramics.

Three dimensional printing

In three dimensional printing, powder is spread over a surface. Using a technology similar to ink-jet printing, a binder material joins particles where the object is to be formed. Following a heat treatment, unbound powder is removed.

Three dimensional plotting

Three dimensional plotting deposits build and support materials. Thermoplastic and wax droplets from 2 ink-jets are printed into a cross section to build a part. A mill passes to obtain the proper height. The final part, a brick of support and build material, is placed into a solvent to remove the support material. The process is also called Sanders prototyping, after the commercially available system.

Ballistic particle deposition

This process works by firing microdroplets of material to form a cross section of the part. A jetting system shoots droplets of molten thermoplastic onto the model surface (10,000 a second). A second heated head ensures a smooth, accurate surface. A material different from the material making up the part can be deposited as support. Afterwards, it is removed in a solvent bath.

Gas Phase Deposition

In Gas Phase Deposition, gas decomposes by heat or light. The shapes grow from the decomposition products in a pattern determined by a laser. It is similar to chemical vapour deposition (CVD), but with a higher deposition rate and selectivity. Gas phase deposition is divided into selective area laser deposition (using a substrate, a reactant gas, and a laser), selective area laser deposition - vapour infiltration (the substrate becomes integral to the shape), and selective laser reactive sintering (the gas and the powder form a new material).

Freeform Powder Moulding

Freeform Powder Moulding fabricates the part from powders, using differences in behaviour of powders when exposed to equal conditions. Under certain conditions, some of the powder (part powder) becomes solid, while the rest of the powder (tool powder) does not. Selectively arranged layers of part and tool powder create a component. Once all layers are deposited, powder processing techniques such as sintering and hot isostatic pressing are used.

Rapid Prototyping by NC milling

This milling approach is based on an STL-file. From the data in this file, milling paths are calculated automatically. No special process is needed; a (small) NC milling machine combined with some special software is sufficient. The software is different from conventional CAM software (the toolpaths generated are also different). The prototypes are milled in materials that are easy to use, such as PUR foam.

A.2 Rapid prototyping in the product creator

Rapid prototyping and manufacturing can be used to support the designer, as depicted in Figure A.1. It results into a new realisator process R^*. The product model is converted into manufacturing job pr^*. As described above, the procedures and materials (m^*) used are different from the ones used in conventional manufacturing, so information concerning the processing operations used (re_p) is less useful. The resulting product (or actually, the prototype) p^* is also different. Evaluation of a design is based mainly on realised form re_f of the prototype. The result of this evaluation ed^* is passed to the designer. Because the layer-additive processes used only have limited restrictions concerning manufacturability of models, resource information rp to the process planner is omitted.

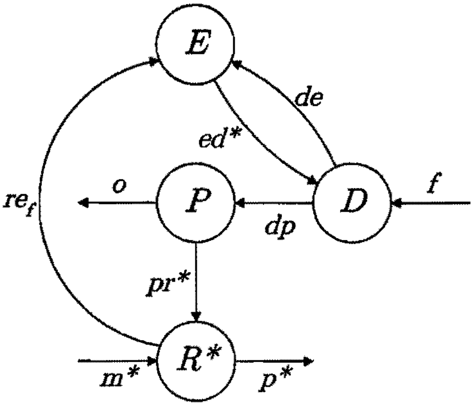


Figure A.1: Rapid prototyping and manufacturing

Appendix B

From function to form

Reasoning from function to form is the essential mode of reasoning in design. A distinction can be made between system-level or conceptual design, embodiment design, and detail design. The main activity at the conceptual or system level design phase is establishing a functional structure. The phase that deals with materialisation of functions is usually called embodiment design. In detail design, a product model is supplied with detailed geometrical information and mating relations between the parts. When reasoning from function to form, product architecture plays an important role [Ulri95]. The product architecture is the scheme by which the functional elements of the product are arranged into the major physical building blocks. These building blocks are made up of a collection of components that implement the functions. A key characteristic of a product architecture is the degree to which it is modular or integral.

In modular architectures, each building block implements a specific set of functional elements and has well-defined interactions with the others. In integral architectures, the implementation of functional elements is spread across building blocks, resulting in ill-defined interactions. A product embodying an integral architecture is often designed with the highest possible performance in mind. Modular approaches require very careful planning during system-level design, while detail design is relatively simple. Integral architectures require less effort in system-level design but need a lot of co-ordination during the detail design phase.

An example of a highly modular design is an Integrated Circuit that is designed using Very Large Scale Integration (VLSI) techniques. Based upon the requirements, the main functions and their logic are designed. The main functions are composed of operations that are described using logic. When the logical structure (architecture) is known, the physical implementation is created by linking library elements. The resulting implementation can be proven correct using Boolean algebra. The design of the components is a separate activity; a library of verified devices with known behaviour is created. The components match exactly one function, so system-level design and component design are not coupled. Such extreme modularity is seldom found in mechanical products, which makes it fundamentally different [Whit96]. In mechanical design, parts participate in or contribute to several functions. The interactions are more complex. Mechanical components do not comply with the rules of logic, because they change behaviour when connected into a system. Consequently, a mechanical system is designed together with its components. Mechanical products are usually not made up by combining library devices.

Suh has proposed that design theory should be approached from the point of interactions. Under this so-called axiomatic design theory, lack of interactions characterises a good design [Suh90]. The problem addressed is divided into four domains, as depicted in Figure B.1. The functions are specified using functional requirements, which are satisfied by design parameters. The relation between functional requirements and design parameters is described using a design matrix A. The design matrix is a diagonal matrix in an uncoupled design. Process variables are related to design parameters by a matrix B. This is used to evaluate manufacturability of a design: matrix A times matrix B must yield a diagonal or triangular matrix [Suh90].

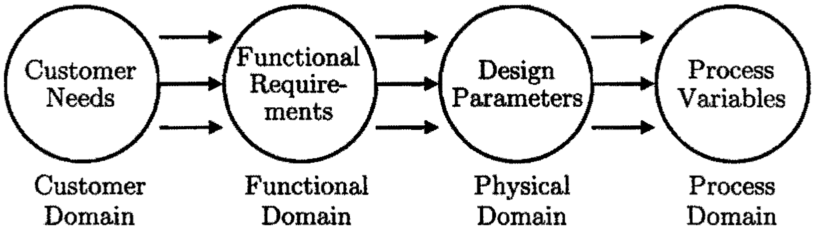


Figure B.1: Design domains in axiomatic design

$$\begin{bmatrix} FR_1 \\ \vdots \\ FR_n \end{bmatrix} = [\, A \,] \bullet \begin{bmatrix} DP_1 \\ \vdots \\ DP_n \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} DP_1 \\ \vdots \\ DP_n \end{bmatrix} = [\, B \,] \bullet \begin{bmatrix} PV_1 \\ \vdots \\ PV_n \end{bmatrix}$$

Axiomatic design attempts to create extremely modular designs. Although most products are modular to a certain degree (because of product variety, component standardisation, manufacturing, or project management) it is not easy, if not impossible, to create completely uncoupled designs. Usually, it is not even recommended to uncouple designs because of performance, suboptimisation and efficiency, although functional coupling should not be confused with physical coupling (more than one function in a single part). This is demonstrated by the fact that VLSI design now faces problems similar to the ones in mechanical design. The use of library elements causes loss of space and the elements made smaller and put closer together. Element design and system design are no longer independent, so the engineer is again needed to design and debug properly [Whit96].

Because modular design enables less complex materialisation of functional requirements, it is a prerequisite if one attempts to automate embodiment design. Attempts are made to automate embodiment design using formal methods, automated reasoning, and libraries of solutions. If many of the steps are done automatically, extremely complex systems can be designed using relatively little effort. Up to now, most attempts in automating design concentrate upon creating models for describing functions and creating a database containing solutions for physical effects and working principles [Tomi89, Kutt93]. The transfer of these solutions to geometry, material properties, and assembly structures is still a problem. The fundamental difficulties lie in the complexity of design knowledge. In a number of cases, collected library elements can not be used in other situations than the originally described one. Domain specific knowledge as well as plain common sense (so-called naive knowledge) has to be modelled. Besides that, the number of library elements in mechanical design knowledge bases becomes enormous if problems exceed toy-problem size [Yosh93].

Automatic embodiment design is limited to products having a modular architecture. Furthermore, the domain is limited and performance of the product is probably less than optimal. Creating a tool that automatically generates a design from functional specifications therefore has limited value. The formal specification of both functions and the design process itself is insufficient at the moment. A large portion of the design process is still left to the human designer. Although reasoning from function to form is the core of designing, it is also the most difficult activity to formalise.

Appendix C

Product modelling

C.1 Relations and primitives

Design primitives

- Halfspaces: infinite faces describing solids in CSG representations [Bron93].
- Features described in [Ambl75]: plane faces, cylindrical shafts or holes.
- Measure entities; physical or artificial elements of geometrical objects used to attach dimensions to a solid component [Sheu93].
- Handles: points on the faces of features that describe the geometry [Cham93].
- Infinite surfaces that were used as primitives in [Kawa93]; planes, cylindrical faces, conical faces, and spheres.
- A structure of features based on boundary modelling, like a cube having subfeatures like a top face or side face [Wood89]. For objects that are not in the boundary model, like a centre-line of a hole, geometric abstractions are used.
- Generic control elements defined in [Shah93]. Planes, axes, and points serve as reference entities and target entities. A reference entity is fixed to the volume being located. Reference-target combinations are created such as parallel, perpendicular, co-planar, and co-axial.
- Associated elementary surfaces for dimensioning and tolerancing: Technologically and Topologically Related Surfaces [Desr95, Salo95]: spherical, plane, cylindrical, helical, and prismatic surfaces, surfaces of revolution, and general surfaces.

Assembly relations

- Against and fits relations to configure the location of parts [Ambl75]. Against relationships define the relative position of planes and cylinders or two planes. Fits relationships describe the relative position of cylindrical planes and shafts.

- Virtual links, describing relationships and mating conditions (against, fits). Every pair of mating components occupies one virtual link, like a rigid attachment, translational constraint, or rotational constraint [Lee85a].

- Mating conditions in [Ko87]: against, fits, tight-fits, and contact. The against condition assumes that two planar faces are in contact; freedom of rotation or slide remains. The fits condition aligns a solid cylinder and a hole, allowing rotation. If a rotation is not allowed, the tight-fits relation is used. Contact relations hold between points of the faces and prevent movements.

- Mating conditions like spherical fits, screw fits, gear contact, and rack and pinion contact [Roch87].

- The against mating condition including surfaces other than planar and the fits mating condition including planar faces [Baxt92]. The contact mating condition is rejected as it has no basis in the physical world.

- Spatial relationships described in [Liu91]: against (planar faces or cylinders), parallel (with an offset), aligned, and the angle between faces.

- Primitive assembly relations described in [Sodh91]: contacts, attachments (contacts enforced by fasteners, weldments or springs), assembly dimensions (constraints that locate an element), enclosures, and alignments.

- The geome concept to model assembly operations and kinematic constraints [Wolt91]: planar-contacts, revolute-constraints, and attached-constraints.

- Assembly conditions as sets of primitive control elements [Shah93]. Constraints are defined between reference and target entities: planes, axes, or points

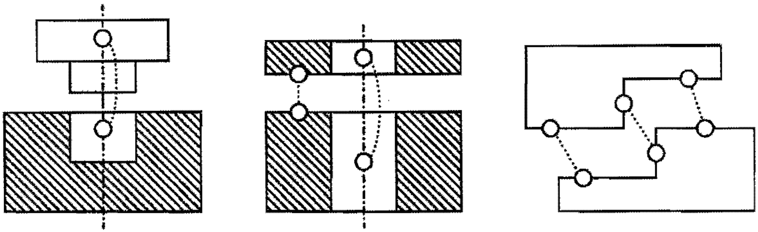
- Mating relations of features like holes, pins, flat planes, and slots [DeFa93]. Feature-models of parts are assembled by indicating feature mates, such as a bore through the axis of a cylinder. Features are provided with assembly information, such as the assembly direction.

Geometrical primitives

- Features; although they use volumes as primitives, they can be studied to find useful functionality.
- An examination of the design of several representative parts. From this, the possibilities that complex primitives should render are found [Coum95].
- The geome concept [Wolt91]. A geome is an arbitrary collection of geometric elements, internally represented by a set of constraints on its elements.
- The design and manufacturing counterparts of the application of one or more tools, machines, and set-ups in the manufacturing phase: manufacturable objects [Delb89]. Manufacturable objects are mapped upon manufacturing operations; they do not contain a pre-defined sequence of manufacturing operations.
- Primitive objects in design that can be mapped upon manufacturing operations [Vrie96]; a cylindrical hole, a wedge, a box, a hollow wedge, and a curved wedge. Besides that, blending, chamfering, and extruding operations are provided.

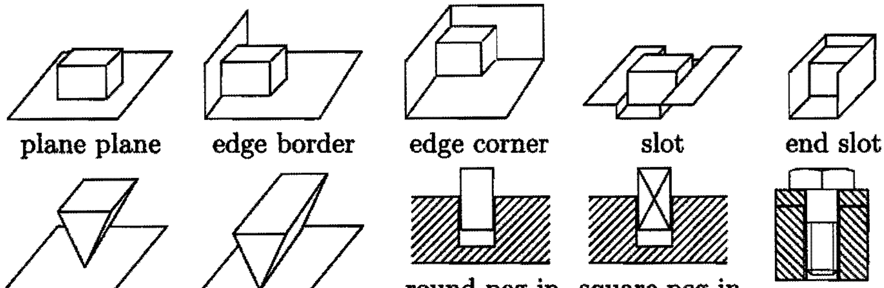
Assembly primitives

- Assembly features, such as a pin joint assembly feature or a mounting assembly feature [Sodh91].
- The sets of relationships to locate part mating frames [Liu91].
- More complex versions of the geome concept, such as a riveted-parts constraint, a two-axis-attachment, or a rack-and-pinion geome [Wolt91].
- German standard DIN 8593: a systematic overview of part connections and assembly processes. It can be used to couple part connections and assembly processes. In this standard, joining operations are divided into nine groups, characterised by the type of bonding and separability of the operations.
- Assembly conditions as a set of generic control elements [Shah93]. Complex relationships are applied between assembly elements, like the insertion feature, alignment feature, or abutment feature depicted in Figure C.1.
- Part relations classified according to the shape of the contact faces [Will97]. The resulting degrees of freedom are an indication of the functional use of a relation. The part relations are depicted in Figure C.2.



insertion alignment abutment

Figure C.1: Definition of assembly features



plane plane edge border edge corner slot end slot

point plane line plane round peg in hole square peg in hole screw

Figure C.2: Part connections

C.2 Propagation

A designer selects reference elements to create relations, so each reference element within a primitive is provided with a unique name. These names relate to the orthogonal directions, local to a primitive, that are identified. The directions are depicted in Figure C.3. The reference elements are named after their location in the primitive, like *top*, *bottom*, *left*, and *right*. Edges are named after the intersection they describe, like *top-left* or *bottom-right*. The names of virtual elements are analogous; *virtual left*, *virtual right*, and so on. It is checked whether the relations do not conflict with the properties of a primitive. For instance, the reference elements of a box should be parallel or perpendicular. Any violation against these internal constraints is reported to the designer, including the nature of the problem.

The propagation method works by counting the number of reference elements that are specified and subjecting them to a series of simple tests, such as a test if the elements are in opposite planes of the primitive [Net94]. This can be represented

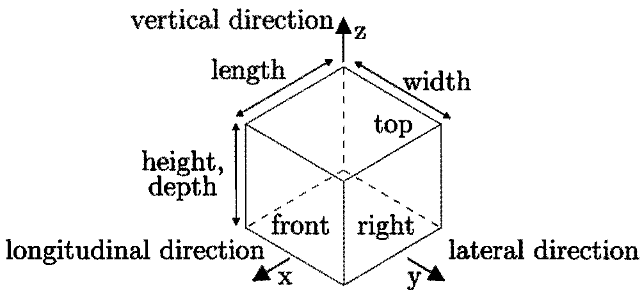


Figure C.3: Conventions of a primitive

as the tree-like structure, depicted in Figure C.4. The procedure also indicates the importance of the first element that is specified. Due to internal constraints, the first element restricts the possible configurations of the following elements.

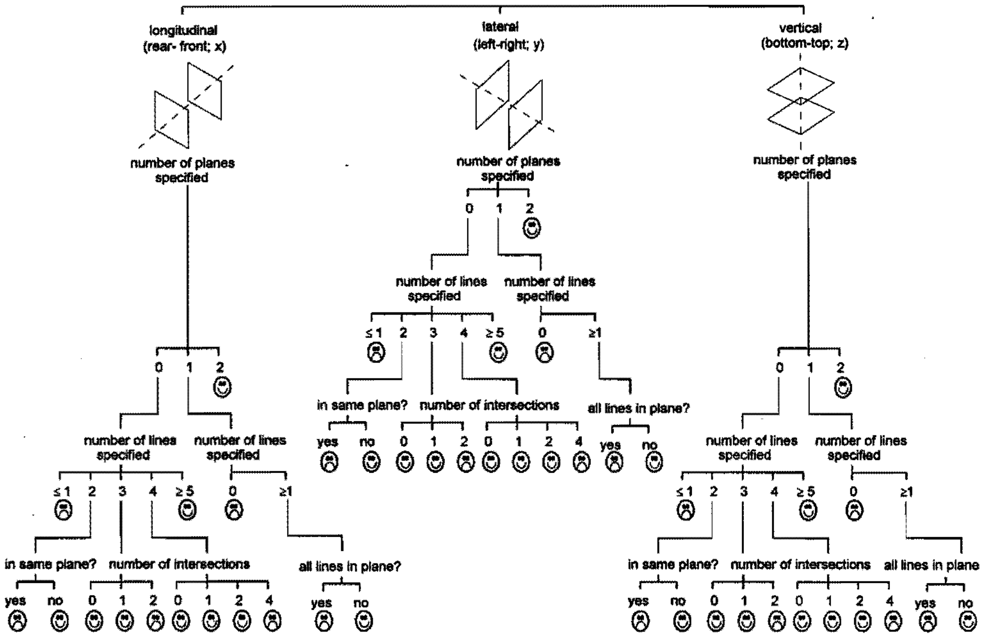


Figure C.4: Propagation of relations

Besides inferring them from relations, the primitive parameter values can also (partly) be specified by the designer directly. The propagation algorithm is able to deal with absolute values as well. If for instance dimensions or locations of primitives are already known, the relations are used to supply the information that is missing. Because of the absolute values provided, these relations will be subject to constraints that add up to the internal constraints [Net94]. Although absolute values are useful in some cases, their functional foundations are doubtful, especially when positioning is concerned. The resulting tree is much simpler, as depicted in Figure C.5.

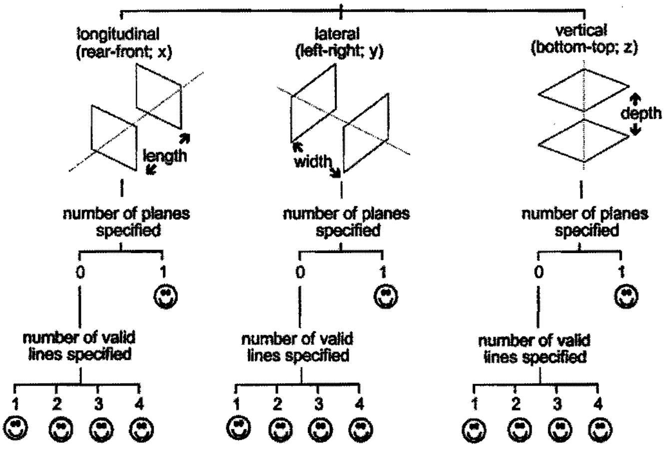


Figure C.5: Propagation for given dimensions

Appendix D

Design grammar

$$\begin{aligned}
\langle Product\ Model\rangle ::= {} & \{\langle Design\ State\rangle\} \\
\langle Design\ State\rangle ::= {} & \{\langle Design\ Substate\rangle\} \\
& [\{\langle Assembly\ Transformation\rangle\}] \\
\langle Design\ Sub-State\rangle ::= {} & \{\langle Geometrical\ Transformation\rangle\} \\
& [\{\langle Material\ Transformation\rangle\}]\ \langle Material\rangle
\end{aligned}$$

$$\begin{aligned}
\langle Assembly\ Transformation\rangle ::= {} & \langle Assembly\ Primitive\rangle \\
\langle Geometrical\ Transformation\rangle ::= {} & \langle Geometrical\ Primitive\rangle \\
& \langle Geometrical\ Operator\rangle \\
\langle Material\ Transformation\rangle ::= {} & \langle Surface\ Treatment\rangle \mid \\
& \langle Material\ Treatment\rangle \\
\langle Material\rangle ::= {} & C35 \mid C45 \mid CuSi2Mn \mid GG20 \mid \ldots \\
\langle Assembly\ Primitive\rangle ::= {} & \langle Assembly\ Primitive\ Type\rangle \\
& \{\langle Relation\rangle\}\ [\{\langle Attribute\rangle\}] \\
\langle Geometrical\ Primitive\rangle ::= {} & \langle Geometrical\ Primitive\ Type\rangle\ \{\langle Relation\rangle\} \\
& [\{\langle Dimension\rangle\}]\ [\{\langle Attribute\rangle\}] \\
\langle Geometrical\ Operator\rangle ::= {} & Add \mid Remove \\
\langle Surface\ Treatment\rangle ::= {} & Coating \mid Painting \mid \ldots\ \langle Surface\rangle \\
\langle Material\ Treatment\rangle ::= {} & Hardening \mid Annealing \mid \ldots
\end{aligned}$$

⟨*Assembly Primitive Type*⟩ ::= *Peg in hole* | *Mounting* | …
⟨*Geometrical Primitive Type*⟩ ::= *Box* | *Cylinder* | *Slot* | …
⟨*Relation*⟩ ::= ⟨*Reference Element*⟩ ⟨*Relation Operator*⟩
[⟨*Dimension*⟩] ⟨*Reference Element*⟩
⟨*Dimension*⟩ ::= ⟨*Name*⟩ [⟨*Number*⟩] [⟨*Unit*⟩]
⟨*Attribute*⟩ ::= ⟨*Lubrication*⟩ | ⟨*Threaded*⟩ | ⟨*Tooltip*⟩ | …
⟨*Surface*⟩ ::= {⟨*Reference Element*⟩}

⟨*Reference Element*⟩ ::= ⟨*Reference Element Type*⟩ ⟨*Virtual*⟩
⟨*Vector*⟩ ⟨*Vertex*⟩ ⟨*Envelope*⟩
⟨*Name*⟩ ::= *length* | *width* | *heigth* | *distance* |
radius | *roughness* | *x* | *y* | *z* | …
⟨*Unit*⟩ ::= *mm* | *μm* | *inch* | mm^{-1} | *rad* | …
⟨*Lubrication*⟩ ::= *Oil* | *Grease* | *Graphite* | *PFTE* | …
⟨*Threaded*⟩ ::= *None* | *ISO Metric* | …
[*M*⟨*Dimension*⟩ × ⟨*Dimension*⟩ −
⟨*Dimension*⟩ (e | *g* | *h*) ⟨*Dimension*⟩
(e | *g* | *h*)]
⟨*Tooltip*⟩ ::= *True* | *False*

⟨*Reference Element Type*⟩ ::= *Plane* | *Cylinder* | *Cone* |
Sphere | *Line*
⟨*Relation Operator*⟩ ::= (*fit* | ◎) | (*against* | ⊻) | (*align* | ⇈) |
(*perpendicular* | ⊥) | (*through plane* | ⊟) |
(*parallel* | //) | (*angle* | ∠)
⟨*Virtual*⟩ ::= *True* | *False*
⟨*Vector*⟩ ::= ⟨*Dimension*⟩ ⟨*Dimension*⟩ ⟨*Dimension*⟩
⟨*Vertex*⟩ ::= ⟨*Dimension*⟩ ⟨*Dimension*⟩ ⟨*Dimension*⟩
⟨*Envelope*⟩ ::= (⟨*Median Face*⟩ ⟨*Dimension*⟩) |
(⟨*Boundary*⟩ ⟨*Boundary*⟩)
⟨*Median Face*⟩ ::= ⟨*Vector*⟩ ⟨*Vertex*⟩
⟨*Boundary*⟩ ::= ⟨*Vector*⟩ ⟨*Vertex*⟩

Glossary

Abstraction	a collection of smaller entities that is perceived as a single entity in a certain domain.
Assembly (1)	the activity that transforms a group of loose parts into an assembled product.
Assembly (2)	a structure consisting of multiple parts.
Basic operation	an operation performing transformations on the material flow.
Complex primitive	a structure composed of primitives.
Design	a creative and analytical process to satisfy a need or solve a problem.
Design intent	the designer's view on the function of a part or product.
Design method	a logical procedure or tool used when designing, such as function analysis or QFD.
Designer	a process creating a product model from functional requirements.
Design model	a model describing design activities, usually presented as a flow diagram.
Evaluator	a process comparing realised properties of a product with the design requirements.
Envelope	a volume describing the limits of size and form of a face.
Feature	an element or property of an object whose presence is relevant for its function, manufacture, design, and so on.
Geometric model	a valid computer based representation of an object's form.
Grammar	the formal specification of a set of primitives and productions to specify transformations of those primitives.
Industrial system	a model comprising the product, the production system, and their mutual relations.
Lead time	the time needed to create a product, including design.
Life-cycle	a description of the status of a system in time.

Manufacturability	the property of a product that describes if operations can be found to manufacture it.
Micro process plan	a description of the manufacturing processes used in a basic operation.
Primitive	the smallest meaningful entity that can be identified.
Process planner	a process performing process planning.
Process planning	the activity that converts product descriptions into working instructions.
Product creator	a process that transforms functional requirements of a customer into products.
Process capability	the performance that is economically possible for a process to deliver.
Product model	a suitable representation of an artifact being designed.
Propagation	calculation of the results of relations that are specified.
Quality	a product's or part's ability to meet the functional requirements, while using minimal resources.
Quality number	a number expressing manufacturing complexity.
Rapid prototyping	the generation of objects directly from a CAD database.
Recipe	a structure describing the basic operations used in creating a product.
Realisator	a process performing manufacturing operations.
Reference element	a primitive used in a relation; a face or line.
State	a representation of space, time, and properties.
Tolerance	a property controlling the stochastic behaviour of the dimensions of a product.
Transformation	a time-dependent action that changes the information in a state.

Curriculum Vitae

Ton van der Net was born on July 17^{th} 1969 in Tilburg, the Netherlands. He attended the Mill Hill College in Goirle, where he obtained his Gymnasium-β diploma in June 1987. In September 1987, he started his study in Mechanical Engineering at Eindhoven University of Technology. During the final years of his study, he worked as an assistant in the laboratory for forming technology, focusing on the measurement of friction in deep drawing. He graduated in the Manufacturing Machines group, where he worked on the design of equipment for manufacturing dialysers. He received his master's degree in March 1993.

Immediately afterwards, he entered a post-masters programme in technological design at the Stan Ackermans Institute; Computer Aided Design and Manufacture of Discrete products. This programme was concluded with a project on the redesign of gas-discharge lamps, aimed at ease of automatic assembly and improved quality. This project was finished in April 1995. During the final year, he was offered the opportunity to start a Ph.D. study on the integration of design and manufacturing. The research started in the Manufacturing Machines group and was continued in the Systems Engineering group, under supervision of Prof.dr.ir. J.E. Rooda. During his Ph.D. study, he served as the coordinator for the post-masters programme Computer Aided Design and Manufacture of Discrete products.

Stellingen

behorende bij het proefschrift

Designing and Manufacturing Assemblies

A.J. van der Net

1. Het terugkoppelen van fabricagekennis naar de ontwerper resulteert in een beter gebruik van toleranties en voorkomt ontwerpfouten. Als gevolg hiervan is zowel een kostenbesparing als een kortere doorlooptijd te realiseren.

   Dit proefschrift.

2. Omdat bij het geometrisch ontwerpen wordt uitgegaan van een scheiding tussen dimensioneren en positioneren, is men niet in staat functionele aspecten goed in een ontwerprepresentatie weer te geven.

   Dit proefschrift.

3. Werktuigbouwkundig ontwerpers zijn over het algemeen niet vertrouwd met formele methoden. Het gebruik ervan zal echter het ontwerpen en fabriceren van producten ten goede komen.

   Dit proefschrift.

4. De tekeningloze fabriek is nog verre van realiteit. Er zijn sinds de Middeleeuwen geen wezenlijke veranderingen opgetreden in representaties van het ontwerp.

   De kaft van dit proefschrift.

5. Technieken als DFA, QFD en FMEA zijn gebaseerd op het gebruik van het boerenverstand.

6. Het begrip *feature* is betekenisloos. De term dient te worden vermeden.

7. Toleranties worden in handboeken vaak behandeld bij de rubriek *werktuigbouwkundig tekenen*. Dit is illustratief voor de wijze waarop met toleranties wordt omgegaan.

8. Spreken over *de* ontwerpmethode is net zoiets als spreken over *de* bank.

9. Carpoolen is een vorm van datacompressie op de snelweg. Anders dan op de electronische snelweg wordt deze mogelijkheid tot het voorkomen van opstoppingen nauwelijks gebruikt.

10. De aandacht die ingenieurs besteden aan het inrichten van een fabriek is niet evenredig met de aandacht die zij besteden aan het inrichten van de eigen werkplek.

11. Omdat mobiele telefoons cadeau worden gedaan bij een wasmiddel, zijn deze niet meer statusverhogend.

12. Lezers van proefschriften zijn verdeeld in twee groepen: zij die stellingen een wezenlijk onderdeel van het proefschrift vinden en zij die stellingen onzinnig vinden. Beide groepen lezen allereerst de stellingen.

Ton van der Net, september 1998.