Mariana Reyes Pérez

A specification technique for the conceptual design of manufacturing systems

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar

Satz und Gestaltung: Mariana Reyes Pérez

Hersteller: Verlagshaus Monsenstein und Vannerdat OHG
Druck · Buch · Verlag
Bielefeld

Printed in Germany

Foreword

The core of our tasks is the product development process. This process ranges from the product idea or business idea to a successful market entry and covers the activities of product planning, of product development and of production system development.

The product and the manufacturing system (resp. production system) have to be developed in a close interplay because the product is strongly influenced by production technologies. This integrative development process must start as soon as first relevant information for production is available, namely during conceptual design.

In our opinion, the principle solution of the product sets the starting point for the conceptual design of the manufacturing system. It is described with a specification technique, which supports the communication and the coordination of the experts across the design process. Such a unified language was needed for the design of manufacturing systems as well.

Considering the above, there is a necessity for a unified language that supports the communication and coordination between the design activities across the entire design process. Such a unified language then provides an enfolding specification for the conceptual design of manufacturing systems at an early stage of the product development process. Against this background, Mrs. Reyes has developed a novel specification technique. It provides a language for the conceptual design and documentation of the principle solution of manufacturing systems at an early stage of the product development process. The work of Mrs. Reyes is a significant contribution to the advancement of design methodology for manufacturing systems. This specification technique for the conceptual design of manufacturing systems supports the interplay between product conception and production system conception. It is another step forward towards our vision of a new school for the design of technical systems of tomorrow.

Paderborn, January 2011 Prof. Dr.-Ing. J. Gausemeier

A specification technique for the conceptual design of manufacturing systems

zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
der Fakultät Maschinenbau
der Universität Paderborn

genehmigte
DISSERTATION

von
Ing. Mariana Reyes Pérez
aus *Mexico City, Mexico*

Tag des Kolloquiums:	22. Dezember 2010
Referent:	Prof. Dr.-Ing. Jürgen Gausemeier
Korreferent:	Prof. Dr. Wilhelm Schäfer
Korreferent:	Prof. Dr.-Ing. Volker Schöppner

Acknowledgements

This work “A specification technique for the conceptual design of manufacturing systems” is the outcome of three years research at the Heinz Nixdorf Institute, University of Paderborn.

First and foremost, I am eternally grateful to Prof. Dr.-Ing. Jürgen Gausemeier for giving me the chance to conduct my doctoral research in his working group. I really appreciate his advices and supervision in the previous years.

I am very grateful for the co-advisorship of Prof. Dr.-Ing. habil. Wilhelm Dangelmaier and Prof. Dr. rer. nat. Wilhelm Schäfer. I thank the International Graduate School of Dynamic Intelligent Systems at the University of Paderborn for its generous fellowship and many other supports.

Regarding my work, I am particularly indebted to my current team leader Dipl.-Wirt.-Ing. Daniel Nordsiek for his dedicated guidance and constructive criticism. Besides that, I am truly grateful to Dipl.-Ing. Rinje Brandis for his valuable inputs with respect to the contents and application example in this work. Special thanks to Dipl.-Wirt.-Ing. Dominic Dettmer for his patience as well as the open and insightful discussions over the last three years.

I would like to thank all current and former colleagues, especially the team “Integrative Production System Design”, for their acceptance and tolerance in all my endeavors in the team. I thank M.Sc. Dipl.-Ing.(FH) Frank Bauer, Dipl.-Ing. Rinje Brandis, Dipl.-Wirt.-Ing. Jan Brökelmann, Dipl.-Wirt.-Ing. Dominic Dettmer, Dipl.-Ing. Johannes Kolsch and Dipl.-Wirt.-Ing. Daniel Nordsiek.

I also have to thank Sabine Illigen and Alexandra Dutschke as well as the team of Dipl.-Ing. Karsten Mette for their kind assistance all the time.

I must acknowledge Dipl.-Lebensmittelchemikerin Angie Martinez and Dipl.-Math. Maria Semenyak who assisted and encouraged me during these three years. Thank you girls for the unconditional support! I am indeed grateful to all friends for their encouragement and motivation.

I thank my family and relatives for their unconditional care, in the case of my parents from the other side of the globe and in the case of my husband from a “neighbour” country. Finalmente, le dedico este trabajo al Señor.

Paderborn, October 2010 Mariana Reyes Pérez

List of Published Partial Results

[RGN09] REYES PEREZ, M.; GAUSEMEIER, J.; NORDSIEK, D.: Ontology Development for a Manufacturing Data Base for Products with Graded Properties. In: International Conference on Information, Process, and Knowledge Management eKNOW 2009, Cancun, Mexico, February 1-6, 2009

[DRW+09] DETTMER, D.; REYES PEREZ, M.; WAGNER, T.; BIERMANN, D.; M.; GAUSEMEIER, J.: A Procedure Model for Manufacturing Process Planning of Products with Graded Properties. In: 3rd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2009), October 05-07, 2009, Munich, Germany

[BGN+09] BRANDIS, R.; GAUSEMEIER, J.; NORDSIEK, D.; REYES-PEREZ, M.: Specification Technique for the Conceptual Design of Production Systems regarding on the Interaction between Product and Production System. In: 3rd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2009), October 5 – 7 2009, Munich, Germany

[RBG09] REYES-PEREZ, M.; BROECKELMANN, J.; GAUSEMEIER, J.: Towards An Expert System For The Manufacturing System Planning Of Products With Graded Properties. In: Proceedings of KEOD 2009 – First International Conference on Knowledge Engineering and Ontology Development (KEOD 2009), October 6-8 2009, Madeira, Portugal

[GBR10] GAUSEMEIER, J.; BRANDIS, R.; REYES-PEREZ, M.: A Specification Technique For The Integrative Conceptual Design Of Mechatronic Products And Production Systems. In: Proceedings of the International Design Conference - DESIGN 2010, May 2010

A specification technique for the conceptual design of manufacturing systems

Table of contents

Page

1 Introduction

1.1 Problem statement

Customer requirements are becoming more demanding. Higher product variety and complexity, shorter product lifecycles and faster delivery times are today's challenges. Companies need continuous product development activities to remain competitive in the marketplace, but, at the same time, they must optimise the resources usage (time, material, etc.) to remain in the market.

With the growing diversification of customers and with the traditional markets going out, every market and product constraint must be considered soft and negotiable. According to FUKUDA [Fuk07], we are between the so called *collaborative engineering*, and a new engineering design generation, where market, product and time constraints are "negotiable". In collaborative engineering the target was to set a strategic goal across the different development processes. To achieve this, communication and collaboration were essential. But now the design problem changed from constraint-driven to goal-driven. The challenge to be afforded now are the constraints. The constraints are now negotiable in number and in priority. This calls for a new paradigm in engineering: communication and negotiation between the product and manufacturing system design. Consequently, a suitable exchange of knowledge and information must take place between product design and manufacturing system design. Among other reasons, this is crucial because the product is strongly influenced by the production technologies. Therefore, the manufacturing system design and the product must be concurrently designed.

To accomplish the product and manufacturing system integration, early coordination between the product and the manufacturing system is required. Along the development flow, the more problems can be resolved during the early development phases, the fewer problems will have to be resolved later by costly investigations based on real prototypes [Trä07]. Furthermore, the parallel design of a product and production system makes possible to reduce the development time, to coordinate design decisions in the early phases and in this way avoid expensive changes in the implementation phase [VDI2206].

A good approach for this early coordination between product and manufacturing system's conception is the use of the product's principle solution as the starting point for the manufacturing system conception. The product's principle solution offers a rough concept of the product. The principle solution is the result of the conceptual design and specifies the basic structure and function of the product. This concept is enough concrete to understand the overall product and its functioning, but still is abstract enough to allow changes, modifications and additions.

The early coordination between the product's principle solution and the manufacturing system requires a **unified language** that supports the communication between the interdisciplinary design team and coordination between the design activities across the entire design process. This is especially important during the conceptual design stage of complex products. Due to the product's complexity, the development results must be specified successively with the development process. The concretization of the design must be described and documented, not only to enable an integrative design but also to support constructive discussions within the interdisciplinary development team.

To facilitate an integrative design, an holistic[1] view must be afforded. An inherent difficulty with this approach is the fact that the task involves experts from different domains. Each domain has its own vocabulary and working methods. This complicates even more the exchange of information.

From an educational and an industrial perspective, providing manufacturing-specific concepts and their integration with product modelling concepts are a matter of special interest. Specialized languages provide suitable terminology for the domain of interest. Therefore, the modeller does not have to reconstruct generic terms into more specific ones. This fact increases the comprehensibility and clarity of the models.

To sum up, there is a gap between the product conceptual design and the manufacturing system conceptual design. A possible bridge between both conceptual designs could be a first rough **description** of the manufacturing system. It should be a basic description which specifies the essential structure and functionality of the system, enough concrete to understand the overall system, but still abstract enough to allow changes and modifications. For this reason, analogue to the product's principle solution, a manufacturing system's principle solution is required. Furthermore, to allow a smooth communication and negotiation between the both principle solutions is advisable to use a graphical modelling language as the unified language. Moreover, a specialized graphical specification technique is advisable, because it provides the suitable terminology, graphics and modelling rules that are necessary for the design.

In a further step, methodologies to manage the transition from the manufacturing system's principle solution to the involved four main production system's aspects *process planning, place of work planning, production logistics* and *working appliance planning,* need to be created.

This specification technique must set up the basis for the exchange of information between product and manufacturing system conceptual designs. Moreover, it must represent the first step and basis for the experts' communication and cooperation in the curse of the system's further concretization.

[1] "***Holism***, *from ὅλος holos, a Greek word meaning "whole". Philosophical theory: the view that a whole system of beliefs must be analysed rather than simply its individual components*." [Enc10-ol]

Today, existing specification techniques do not provide a suitable platform for the conceptual design of a manufacturing system at an early stage of the product development process. Furthermore, different views and backgrounds of the system's developers summed to the inherent complexity of manufacturing systems enforce the use of many models through different stages and phases of the lifecycle of a product [SJ08][ST97]. The aimed representation must fully describe the system but yet be manageable. Therefore, the development of a specification technique for the conceptual design of manufacturing systems at an early stage of the product development process is required.

1.2 Objective

The goal of this work is to provide a language for the conceptual design, development, description and documentation of the principle solution of manufacturing systems at an early stage of the product development process. To achieve this goal, an holistic and integrative[2] specification technique is going to be developed. The aimed manufacturing system's specification technique must have the following characteristics:

- It must provide a principle approach of the conceptual design of manufacturing systems. This principle approach must include the starting point for the integrative manufacturing system conceptual design and how to represent a first manufacturing system concept with different levels of detail and abstraction
- It must be an holistic and integrative semi-formal specification technique
- Enable a collaborative development
- Enable the conceptual design, development, and discussion of the manufacturing system's principle solution within an interdisciplinary team
- Enable a interdisciplinary visualization, representation and documentation of a manufacturing system's principle solution by means of a model, including its structure and general function
- Enable an effective exchange of information and knowledge
- A balanced consideration between the involved domains
- It must provide a simple, common, consistent, graphic language. In this way it supports a consistent terminology and enables an intuitive work

This specification technique should provide a suitable information base for the further concretization of the manufacturing system. One possible way could be the system's concretization by means of methodologies that manage the transition from the manufac-

[2] The term integrative is used because the specification considers as starting point the product's principle solution. In this way the product is considered within the manufacturing system conception.

turing system's principle solution to the involved four main production system's aspects: *Process planning, place of work planning, production logistics* and *working appliance planning*.

1.3 Approach

In the following paragraphs, the structure of the work is going to be presented.

Chapter 2 starts with the definition of terms. It is followed by the principles of product design and manufacturing system design. In this section, the product development and its principle solution are explained. Afterwards, the manufacturing system development is discussed, including themes like design methodologies and models, as well as the contents of the manufacturing system's principle solution. After defining the problems to be resolved, the requirements for a specification technique for the conceptual design of manufacturing systems are outlined.

Chapter 3 analyses the state of the art for this work. It covers domain-spanning design methodologies and specification techniques for systems engineering, production and manufacturing. At the end of the chapter, the need for action is described.

Chapter 4 starts with an explanation of the work's basic idea: A specification based on a coherent system of cross-linked partial models. Further, it describes a principle approach of the conceptual design of manufacturing systems. After that, the main elements of the specification are introduced, including its constructs, pictograms and relationships, as well as its partial models. Subsequently, the specification of the partial models' cross-linking is elucidated. Additionally, the specification's modelling rules and data model are explained. At the end of the chapter, the evaluation of the specification against the requirements outlined in Chapter 2 is presented.

Chapter 5 presents a brief introduction into the application example. In this chapter, the specification presented in Chapter 4 is exemplified using as demonstrator a miniature robot.

Chapter 6 summarizes the work done and subsequently proposes the future work.

2 Problem analysis

2.1 Definition of terminologies

Manufacturing system design integrates diverse actors and approaches. These actors have diverse backgrounds (business, economics, engineering), disciplines (mechanics, electronics, etc.) and terminologies. The approaches range from strategy to detailed process engineering. This diversity causes, among other things, semantic problems. For this reason, this chapter focuses on relevant definitions and concepts for the manufacturing system design. The following paragraphs state an initial basis for the reader in which the applied terminology of this work is clarified. Starting from language concepts used for the model's representation, this chapter treats issues like systems, design, systems design, and finally, about the manufacturing system design.

Language concepts

The manufacturing system design involves experts from different disciplines. Each discipline has its own vocabulary and working methods. For this reason a common language is necessary to ensure the correct information exchange. Specification techniques are a type of language that can be used as communication medium.

Different types of specification techniques are implemented in the course of the manufacturing system design. These specifications are used for the development, description and documentation of the process design. Commonly, they have a pre-defined semantic and syntax. The **semantic** describes the meanings of the used signs, the relation between them and what the signs mean, what they refer to. The **syntax** defines the representation of the language's components. Specification techniques can be classified into three types, according its formalization degree:

Informal specification techniques do not have specific constructs, modelling rules, syntax or semantic. These specifications are easy to understand; nevertheless they are not adequate for the technical description, because they are not precise enough and may induce to errors and ambiguities. Sketches and text in prose are some examples of informal specifications [GEK01, Pg. 306] [GF99, Pg. 333f.] [Sch00, Pg. 14f.].

Formal specification techniques have well defined rules, syntax and semantic. These specifications are similar to programming languages. Its modelling is complex. Furthermore, its reading and understanding requires specific knowledge about the language. These specification techniques are used by different types of experts, which have a deep understanding of the language and theme. Program code is an example of a formal specification technique [GEK01, Pg. 306] [GF99, Pg. 333].

Semi-formal specification techniques are a type of graphical modelling language with a predefined semantic and syntax. It comprises the formal definition of the language's symbols and constructs as well as the principles and rules for combining these constructs. The language is not so precise as formal specifications, but also not so vague as informal specification techniques [GFD+08] [GEK01] [GF99] [Sch00].

System concepts

There are many definitions of a system. The International Council on Systems Engineering (INCOSE) deploys the following consensus about the definition of a system**.**

> „***System*** *is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that is, all things required to produce systems-level results. The results include system level qualities, properties, characteristics, functions, behaviour and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected*" [INC10-ol].

As pointed out above, the value added by the system as a whole depends on how the parts are interconnected. To enable the realization of successful systems, an interdisciplinary engineering approach is needed. This approach is known as systems engineering.

> „***Systems Engineering*** *is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation"* [INC10a-ol].

Systems fulfil specific characteristics. According to LINK, the basic characteristics of a system are [Lin01]:

- Systems accomplish a purpose.
- Systems have a hierarchical nature and can be divided into subsystems
- Systems are not static. They must be designed and improved over time until the system is phased out

The value added of the system needs to be assured, during the **system design** process. System design is a process which, starting from the requirements, leads to a concretization of a system. This concretization is expressed in elements and the interaction of these elements.

On the one hand**, design** *can be seen as the conceiving of a whole, a solution concept, the identifying or finding of the required solution elements. It is the intellectual, model-based joining together and connecting of these elements to form a workable whole*

[DH02, Pg. 158]. On the other hand, the **conceptual design** of technical systems can be seen as a part of the design where the basic concept of the system is defined. This term can be understood as follows:

> *„**Conceptual design** is the part of the design process where -by identifying the essential problems through abstraction, establishing function structures, searching for appropriate working principles and combining these into a working structure- the basic solution path is laid down through the elaboration of a solution principle. Conceptual design specifies the principle solution“ [PBF+07, Pg. 159].*

The outcome of conceptual design is the principle solution of the system to be developed. The **principle solution** is the fundamental solution path for a product or system, specifying the basic aspects of the physical operation, the components, and their arrangement but without defining them in detail [PBF+07].

As pointed out in the definitions above, different design phases are involved during the technical systems' design. A type of technical system is a manufacturing system. A **manufacturing system**[3] is a collection of interrelated **resources** that perform a set of activities (**processes**) converting goods into a variety of **products**, services and information in the most possible cost effective way [adapted from GA09].

Manufacturing systems design must holistically integrate the elements of a manufacturing system into a functioning whole. It starts with its requirements and finishes with the operation of the system. According to GAUSEMEIER, the four main aspects of the manufacturing system development are: process planning, place of work planning, production logistics and working appliance planning [GPW09].

In this work a specification technique for the conceptual design of manufacturing systems is created. The **manufacturing system conceptual design** specifies the **manufacturing system principle solution.** The principle solution is the first concept of the system, where the basic structure and operation of the system is established. The principle solution offers a rough concept of a system's concept. This concept is enough concrete to understand the overall system and its functioning, but still is abstract enough to allow changes, modifications and additions. It is the starting point for the further discipline specific concretization of the system.

---

[3] There is no generally shared definition of manufacturing systems (see Chapter 2.2.2.7). For some authors manufacturing systems and production systems are synonyms while for some authors production systems include the manufacturing system along with functions like product development and marketing. The boundary of the system is not always included in the definitions. For this reason, when the term is applied, the corresponding author's definition is also presented. For the cases in which no definition is enclosed, the above definition is meant.

2.2 Principles of product design

The manufacturing system's goal is to produce products. According to GAUSEMEIER, the product development process ranges from the product idea or business idea to a successful market entry and covers the activities strategic product planning, product development and production system development [GPW09]. The product development process cannot be seen as a stringent sequence of process steps. It is more an interplay of tasks that can be structured into three cycles (Figure 2-1).

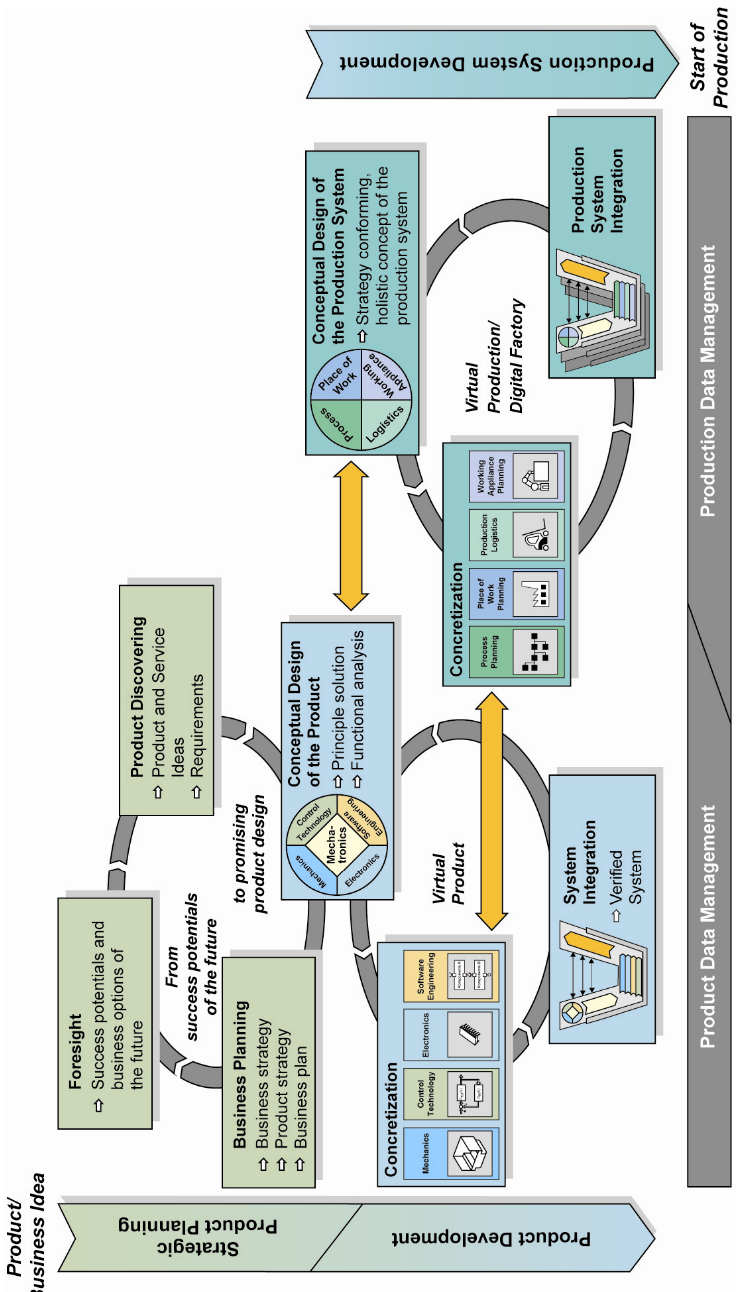

Figure 2-1: Three-cycle-model of product design according to GAUSEMEIER

First cycle: Strategic Product Planning

This cycle characterizes the steps from finding the success potentials of the future to create a promising product design, what in this work is called **principle solution** (see Chapter 2.1). There are the four tasks in this cycle: *Foresight, product discovering, business planning* and *conceptual design*.

Second cycle: Product Development

This cycle covers *conceptual design*, domain-specific *concretization* and development as well as the *system integration*. In this context the creation and analysis of computer models are an important part, which leads to the widely used term Virtual Product and Virtual Prototyping respectively.

Third cycle: Production System Development[4]

The starting point of this cycle is the *conceptual design of the production system*. The result of the conceptual design is the principle solution of the production system. In the further course of the third cycle the four aspects *process planning, place of work planning, production logistics* and *working appliance planning* have to be concretized further.

The product development cannot be seen as a stringent sequence of phases and milestones. The product and the production system have to be developed in a close interplay. This is in particular necessary for complex products, whose concept is affected by the considered manufacturing technology. Because of this, the product development and the production system development are arranged in parallel. This corresponds with concurrent engineering approaches. But unlike them, the close development interplay is already applied in the early design stage, namely the conceptual design. The horizontal arrows in the model of three cycles emphasize this closely interplay. In this context, the principle solution of the product constitutes the starting point for the production system's conceptual design. To understand better this interplay between product and production system, their development is going to be described in more detail in the following chapters.

2.2.1 Product development

Innovative products require an interdisciplinary work from experts from different fields. These experts use different methodologies and languages. Furthermore, they could have different approaches for solving the same problem. To assure a synergistic work it is required an **essential basis** which facilitates communication and cooperation between

[4] Production system development means manufacturing planning (synonym work planning, planning of the bill-of-activities, bill-of operations) complemented by production logistics with a special focus on material flow planning [GPW09].

the experts and their work. Firstly, a common systematic for the product development is required, and secondly, an **adequate language** for the conceptual design, development, description and documentation of the product development is need.

The following sections are intended to give a comprehensive overview of this *essential basis*. At first, an interdisciplinary systematic for the product development, the VDI2206, is going to be presented and then, an adequate language for the conceptual design is going to be explained. This language is a specification technique used for the description of the product's principle solution within interdisciplinary teams.

2.2.1.1 VDI Guideline 2206 (VDI 2206)

The VDI[5] 2206 offers a practical guideline for the systematic development of mechatronic products. The guideline promotes interdisciplinary cooperation, and offers a basis for the cooperative product's development.

Mechatronic products offer potential for success; nevertheless such complex products impose special requirements on the development process. These systems have a high level of complexity, attributable to the greater number of coupled elements which are realized in different technical disciplines (heterogeneity). Furthermore, this type of product requires the integrated combination of different disciplines. To support the development of mechatronic systems, the guideline proposes a flexible generic procedural model for the system's design, the V-model (Figure 2-2).

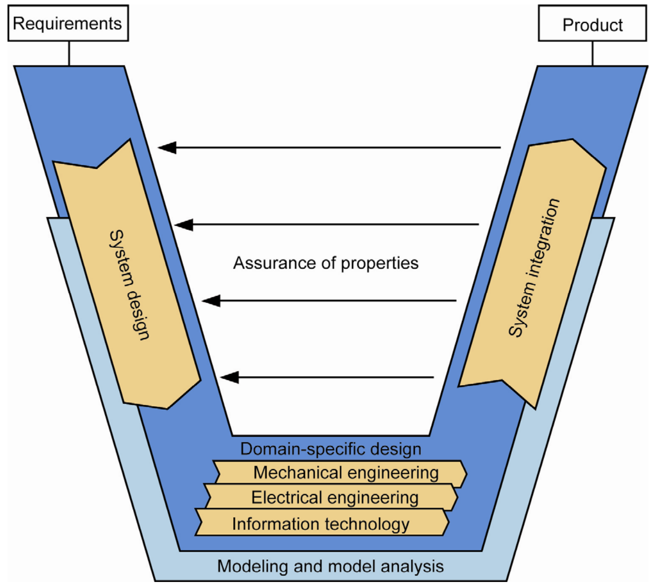

Figure 2-2: The V-Model as a macro-cycle that describes the generic procedure for designing mechatronic systems [VDI2206 Pg. 29]

[5] Verein Deutscher Ingenieure (VDI)

Requirements are derived from the development order. The requirements specify the object and serve at the same time as the measure against which the later product is to be assessed. During the system design phase and based on the previous defined requirements, it is developed a **cross-domain solution concept** which describes the main physical and logical operating characteristics of the future product. To achieve this description, it is necessary to break down the overall function of the system into sub-functions. The hierarchical structure of these sub-functions creates a first functional hierarchy. Then each sub-function is assigned to suitable **operating principles**[6]**.** Based on this information and using the synergies from the interaction of the involved domains, a first **active structure** is derived [FGK+04] [VDI2206] [PB97].

The active structure presents the linkage of operating principles via material, energy and information flows. Although the active structure presents a general concept of the system, is still insufficiently concrete to allow the **principle solution** to be assessed. The active structure must be, amongst other things, quantified and embodied. The further concretization of the active structure leads to the **building structure**[7] [VDI2206] [PB97] and the **component structure**[8]. Shape related system elements are joined together, according to its spatial interrelations and restrictions, to conform the building structure. In contrast, the component structure is the result of the information system elements' concretization to software components.

Once that a first **cross-domain solution concept** is available, further concretization takes place in the domains involved. Then, the results of the individual domains are joined within the **system integration phase**. This integration allows investigating the interaction between the individual domains. Each progress made within the design must be continually checked, to ensure that the actual system properties coincide with the desired system properties (assurance of properties**,** see Figure 2-2). The final result of the system integration is the **end product.**

2.2.1.2 Product's principle solution

A specification technique for the domain-spanning description of the principle solution of self-optimising mechatronic systems has been developed at the Heinz Nixdorf Institute [GFD+09]. This specification was further developed for products in general [GBR10]. Additionally, methodologies are being created to manage the transition from

[6] "*The operating principle refers to the interrelationship between physical effect and geometrical and material-related features (effective geometry, effective movement and material). It allows the principle of the solution for performing a subfunction to be identified", [PB97] [VDI2206, Pg.118].*

[7] *"The building structure takes into consideration the spatial interrelationships and the requirements of production, assembly, etc. defined in it are installation spaces, later components and subassemblies with their associated connections, which represent the concrete product", [PB97] [VDI2206,Pg.34].*

[8] With *component structure* are meant software elements that will be stored/saved/burned within "hardware" components.

the principle solution (domain-spanning description) to the involved disciplines (domain-specific description), please refer to [HHK+08] [GGS+07] [GSG+09] [Che09] for more information.

The mentioned specification technique divided the product's principle solution into aspects. Those aspects are: *Environment, application scenarios, requirements, functions, active structure, shape* and *behaviour* (Figure 2-3). The mentioned aspects are mapped on computer by partial models. Because the aspects are in relationship with each other and ought to form a coherent system, the principle solution consists of a coherent system of partial models. It is necessary to work alternately on the aspects and the according partial models, although there is a certain order [GFD+09].

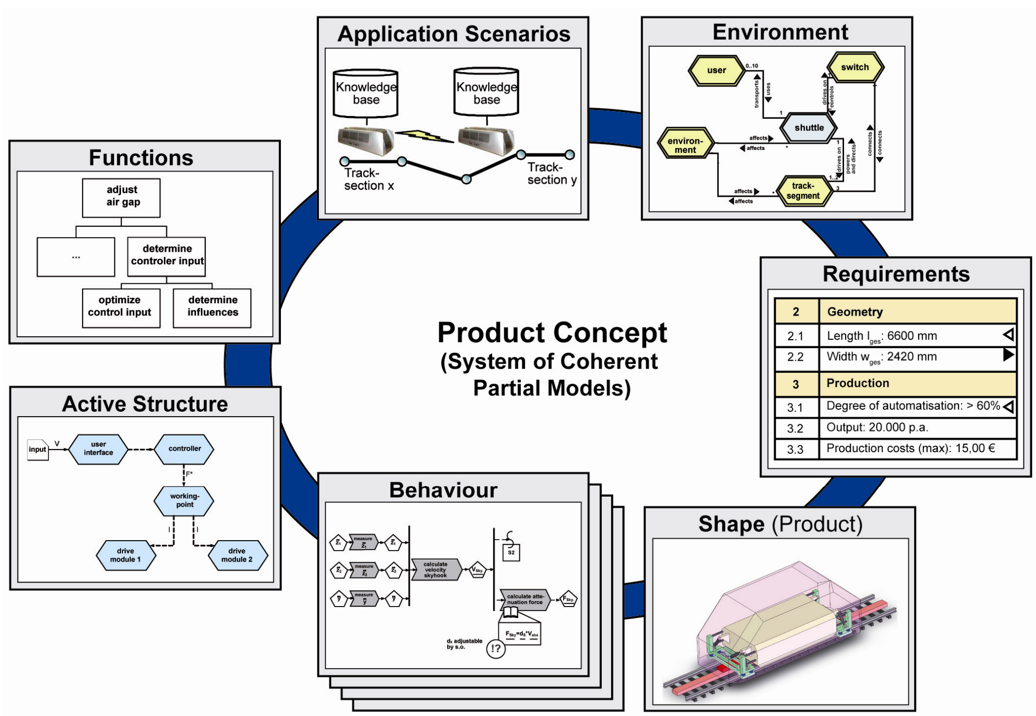

Figure 2-3: Partial models for the domain-spanning description of a product's principle solution, adapted from [GBR10]

The description of the environment, the application scenarios and requirements serve as the starting point. They are usually followed by the function hierarchy and the active structure. The active structure represents the core of the principle solution in conventional mechanical engineering. The mentioned partial models are described below. For a detailed description, please refer to [GFD+09].

Environment: This model describes the environment of the system that has to be developed and its embedding into the environment. Relevant spheres of influence (such as weather, mechanical load, superior systems) and influences (such as thermal radiation, wind energy, information) are identified. Disturbing influences on the system are marked as disturbance variables. Influences that cause a state transition of the system are marked as events. Finally, catalogues, that imply spheres of influences and influences, support the creation of environment models.

Application scenarios: Application scenarios form first concretizations of the system. They concretize the behaviour of the system in a special state and a special situation and furthermore, what kinds of events initiate a certain state transition. Application scenarios characterise a problem, which needs to be solved in special cases, and also roughly describe the possible solution.

Requirements: This partial model presents an organised collection of requirements that need to be fulfilled during the product development (such as overall size, performance data). Requirements are distinguish between functional and non-functional requirements. Functional requirements describe the desired functionality of the system. Non-functional requirements describe properties of the system itself as well as of its behaviour. Every requirement is textually described and, if possible, concretized by attributes and their characteristics. There are checklists that assist the setting up of requirements, for example [PBF+07], [Rot00].

Functions: This partial model concerns the hierarchical subdivision of the functionality. A function is the general and required coherence between input and output parameters, aiming at fulfilling a task. For the setting up of function hierarchies, there are catalogues with functions (e.g. [Lan00]). Functions are realized by solution patterns and their concretizations. A subdivision into sub functions is taking place until useful solution patterns have been found for the functions.

Active structure: The active structure describes the system elements, their attributes as well as the relation of the system elements. Each system element can be decomposed in subsystems. A system element represents a subsystem, a module, a component or a software component. In the case of an environment's system element, the element could be a person/user [ADG+08].

Shape: This partial model describes first definitions of the system's shape. This especially concerns working surfaces, working places, surfaces and frames. The computer-aided modelling takes place by using 3D CAD systems.

Behaviour: Two partial models are used to specify the behaviour of the system. These are the partial models *behaviour – states* and *behaviour – activities*. The partial model *behaviour – states* defines the states of the system and the state transitions. The state transitions describe the reactive behaviour of the system towards incoming events. The partial model *behaviour – activities* describes the logical sequence of activities in the system. Especially, parallel executed activities and their synchronization can be described in this way.

2.2.2 Manufacturing system development

There is no generally shared definition of manufacturing systems. For some authors manufacturing systems and production systems are synonyms while for some authors production systems include the manufacturing system along with functions like product development and marketing (see Chapter 2.2.2.7). The boundary of the system is not always included in the definitions, mainly because the system design and development incorporates numerous research disciplines, involving from manufacturing strategy to detailed process engineering. For this reason, the following chapter treats indistinctly both terms, manufacturing system and production system. The chapter begins with a review of existing frameworks and methodologies for the manufacturing and production system's development. These are reviewed and related with the early stages of the product development process. Based on this review, the chapter concludes with the manufacturing system's principle solution. This definition states the initial point for the here developed specification technique (see Chapter 4).

2.2.2.1 V-Model for the production system

The VDI 2206 presents a practical guideline for the systematic development of mechatronic products and its production system development. In this way, the dependencies between product and production system can be adequately considered. For this reason, additional to the product development model (Figure 2-2), the VDI 2206 presents an analogue model for the production system development (Figure 2-4) [VDI2206].

In the V-Model for the mechatronic systems development, the overall task is subdivided into sub-tasks. These sub-tasks are solved, concretized in each domain and later are brought together again to solve the overall task. Close meshing of product and production system design must be ensured in particular in the early phases of **preparation** and **conception of production** (See Figure 2-4). During the **conception of production** it must be checked whether the necessary production technologies are available and the product can be cost-effective produced. In this phase is where the **planning of the production principle** is done. The result of this planning is the principle solution, which is further concretized within the **domain-specific design** task. The results of this task are sent to the next task, the **planning of technology chains**. Afterwards, one technology chain is going to be concretized with specific resources, within the **detailed and implementation planning** phase. The design activities are supported in all phases by auxiliary means of virtual production/digital factory, including 3D-CAD, FEM, material flow simulation, etc. [VDI2206].

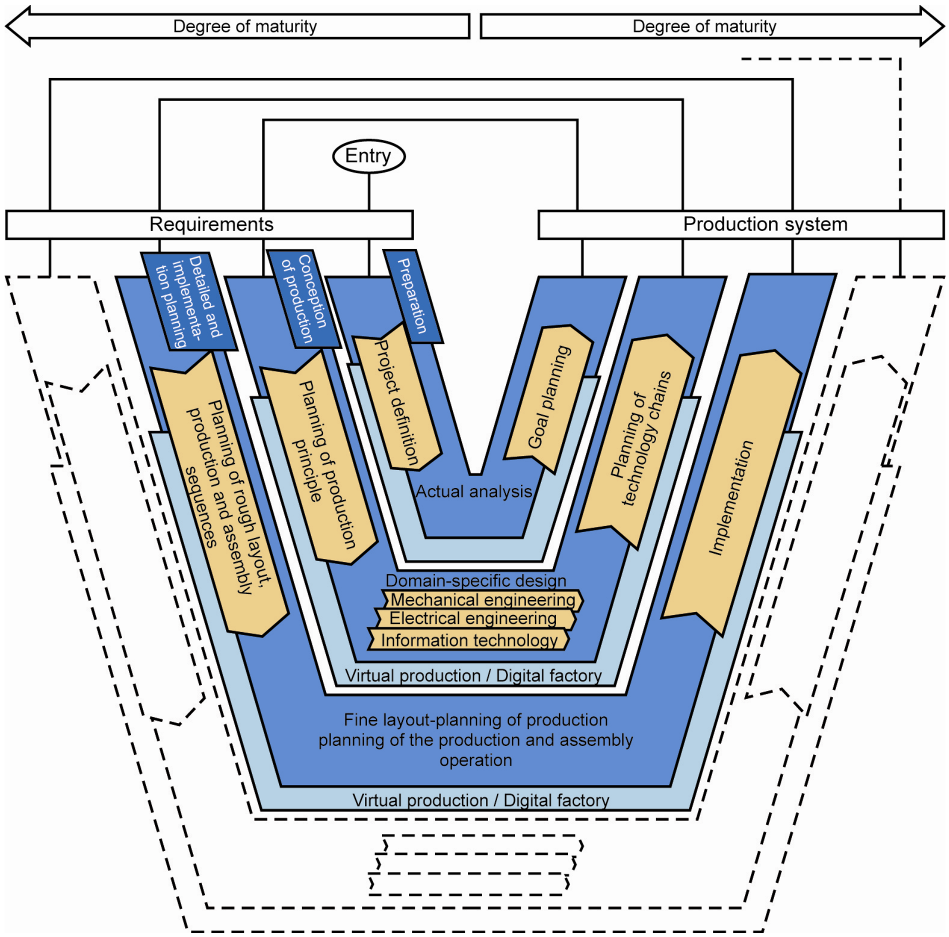

Figure 2-4: The V-Model as a macro-cycle that describes the generic procedure for designing production systems [VDI2206, Pg. 45]

2.2.2.2 GRAI integrated methodology (GIM)

GIM is based on a manufacturing system reference architecture. A reference architecture is a set of models which describe how a system function and its elements. Applied to the manufacturing system context, a reference architecture describes the manufacturing system's elements and functions with the aid of models [Lin01] [WBB+94].

The GIM methodology uses as starting point the GRAI (Graphe a Resultats et Activites Interlies) methodology. GRAI methodology was made by the GRAI/LAP (Laboratory of Automation and Productics) of the University of Bordeaux I. Originally, it was developed to design production management systems. It uses two graphic tools, the GRAIgrid and the GRAInet. The GRAIgrid is a tool designed for the analysis of the production management systems within a manufacturing operation. With this tool is possible to identify what decision centres are required for a coordinated system. The GRAInet is used to describe the activities in each decision centre [Wu92].

The GIM methodology has 6 elements [CVD97]:

1) GRAI conceptual model

   This conceptual model divides a manufacturing system in three sub-systems: Physical system, decision system and information system. The first system describes the elements of the system that transform raw material into products. The second system controls the physical system. Finally, the information system is in charge of giving the required information to the decision system for the decision-making.

2) GIM modelling framework

   It defines the enterprise modelling space with three dimensions: View point axis, lifecycle axis and abstraction level axis.

3) GZM reference architecture

   It is defined as a structured set of ‘models’ which are the building blocks of the manufacturing system.

4) GZM modelling formalisms

   A formalism is a type of language used to represent particular parts/ concepts/ elements of the manufacturing system. The following languages are used: GRAI grid, GRAI nets, IDEF0, S/R and entity/relationship diagrams.

5) GIM structured approach

   Phases used for the analysis and design of a manufacturing system. It contains three main phases: analysis, user oriented design, technical oriented design (Figure 2-6).

6) GIM case tool

   A tool, ‘ PROGRAI’ was developed by GRAI/LAP on Macintosh environment to support the use of GIM.

GIM methodology offers a general description of a manufacturing system, specifically on details in manufacturing control system. It was designed for the conceptual design stage of manufacturing system life cycle. It is useful as reference, guideline, or training tool at the early stages of the system development [Lin01][CVD97][Wu92].

Figure 2-5 presents the phases used in GIM for the analysis and design of a manufacturing system. In the GIM approach, a basic assumption is to start with the analysis of the existing system. Even if the system is *new,* there is always an existent situation, like similar systems, knowledge about the organisation or examples of similar products.

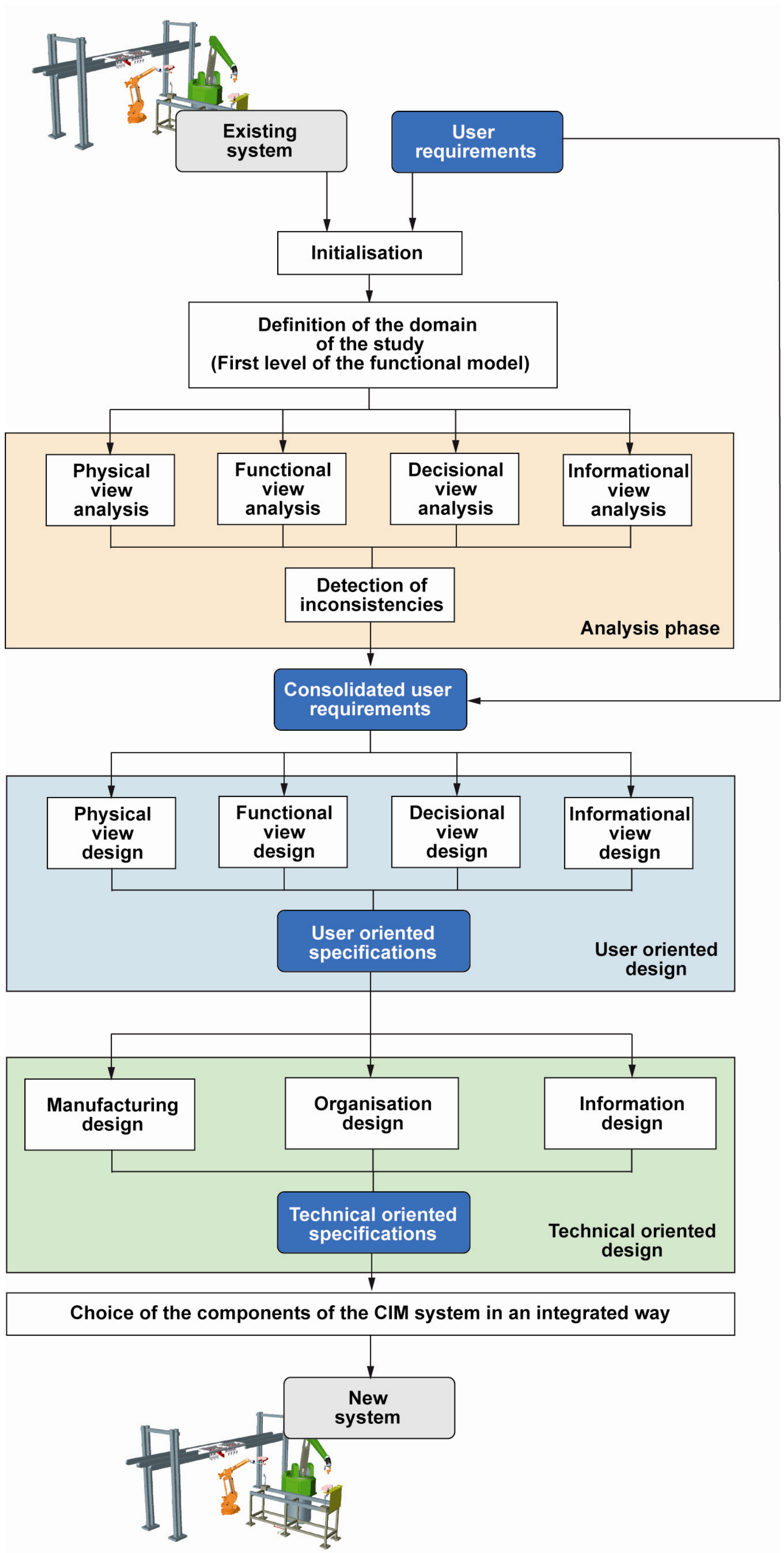

Figure 2-5: GIM structured approach [CVD97]

GIM's structured approach consists of four phases: initialisation, analysis, design, and implementation. Initialisation consists of defining company objectives, the domain of the study, the resources involved, etc. The analysis phase results in the definition of the characteristics of the existing system in terms of four user-oriented views (physical, functional, decisional, informational). The design phase is performed in two stages: user oriented design and technical oriented design. User-oriented design uses the results of the analysis phase to establish requirements for the new system. Technical-oriented design consists of transforming the user-oriented specifications models of the new system design to technical-oriented specifications. These specifications express the system requirements in terms of the required organisation design, information design and manufacturing design. Finally, the new system is implemented [Lin01] [CVD97] [Wu92].

2.2.2.3 Axiomatic Design

Axiomatic Design (AD) is a theory developed by SUH for the development of various kinds of systems such as mechanical systems, software systems and manufacturing systems. In the last case, the AD approach proposes simultaneous consideration of process during the product design. The design methodology gets its name from the use of design principles or design axioms that lead the analysis and decision making process.

According to SUH, design is the interaction between what to achieve and how to achieve in four design domains: **customer**, **functional, physical**, and **process** (Figure 2-6). SUH formally models the domains and the relationships between them [Suh01].

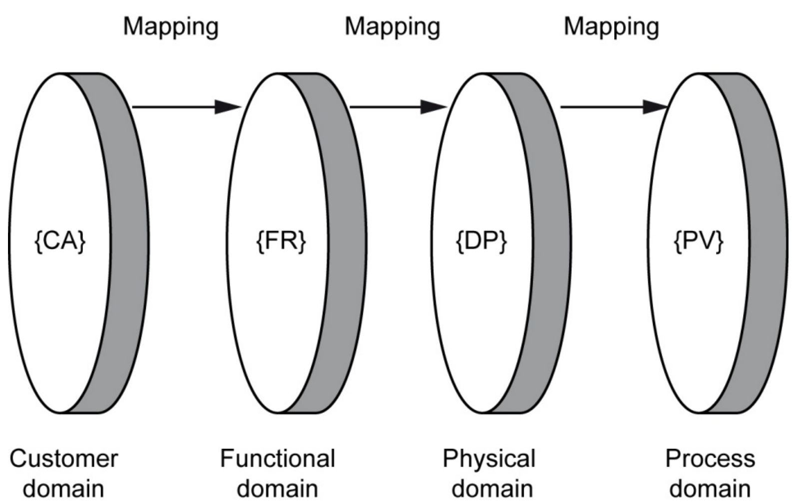

Figure 2-6: Design process of Axiomatic Design, [Suh01, Pg.11]

The design process of AD can be summarized as follows: The customer attributes or needs (CAs) are mapped to functional requirements (FRs), which are satisfied by design parameters (DPs). DPs are physically implemented by process variables (PVs), which are typically quantities such as temperature, pressure, and flow rate [Suh01].

The customer domain is where the *requirements of the system* are specified. In the functional domain, these requirements are concretized into *functional requirements* and *constraints*. The functional requirements represent the functions of the system. In the physical domain, these functional requirements are converted into *design parameters*. Design parameters describe the *physical characteristics* of the system to be developed. The process domain describes the manufacturing process of the system by means of *process parameters*. The parameters involved in a domain are graphically represented by hierarchical trees and mathematically represented by vectors. The transition from one domain into another is specified by *design matrices* [Suh01, p. 18] and the *zigzagging principle*. Design matrices determine how the variables of a domain are transformed into the variables of another domain and the zigzagging principle (see Figure 2-7) guides the decomposition process from a high system level to lower detailed levels [Suh01].

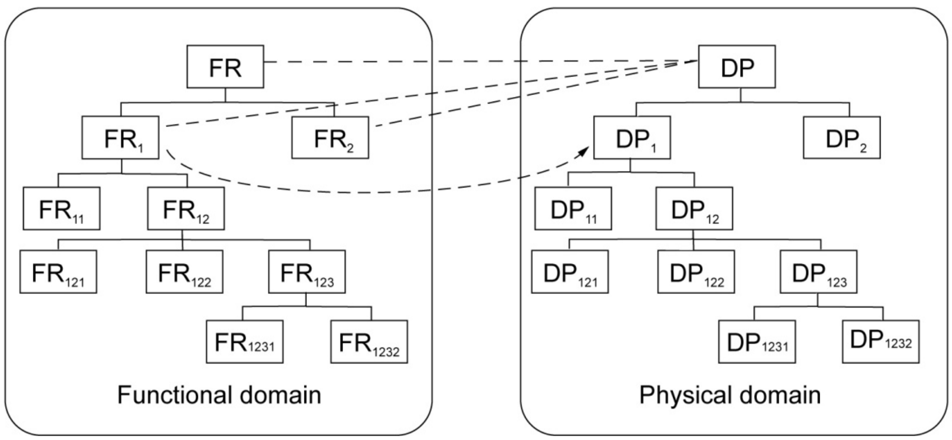

Figure 2-7: Zigzagging to decompose functional requirements (FRs) and design parameters (DPs) in the functional and physical domains, to create the FR and DP hierarchies, adapted from [Suh01, Pg.30]

During the design, there are two fundamental axioms that govern the process. The first axiom is the *independence axiom*. The independence axiom indicates that the independence of functional requirements (FRs) must be maintained. This means that design decisions must be made without breaking the independence of each functional requirement from other functional requirements. The functional requirements must be independent to each other and their number must be minimized to be just enough to characterise the design [Suh01].

The second design axiom is the *information axiom*. Information axiom states to minimize the **information content** of the design. In other words, the minimum information content is the best design. Axiomatic design defines the *information content* as the log inverse of probability of success to satisfy the functional requirements. Based on the two axioms, theorems and corollaries are derived [Suh90]. Many Axiomatic Design applications have appeared in the literature in the last years, some of them can be found in [HJ02][SCP98] [Suh01].

2.2.2.4 Systematic for the planning and implementation of complex production systems

REFA Bundesverband e. V. is a german organisation that deals with work design, companies' organisation and development. This organisation created a systematic for the planning and implementation of complex production systems. Figure 2-8 presents the involved phases.

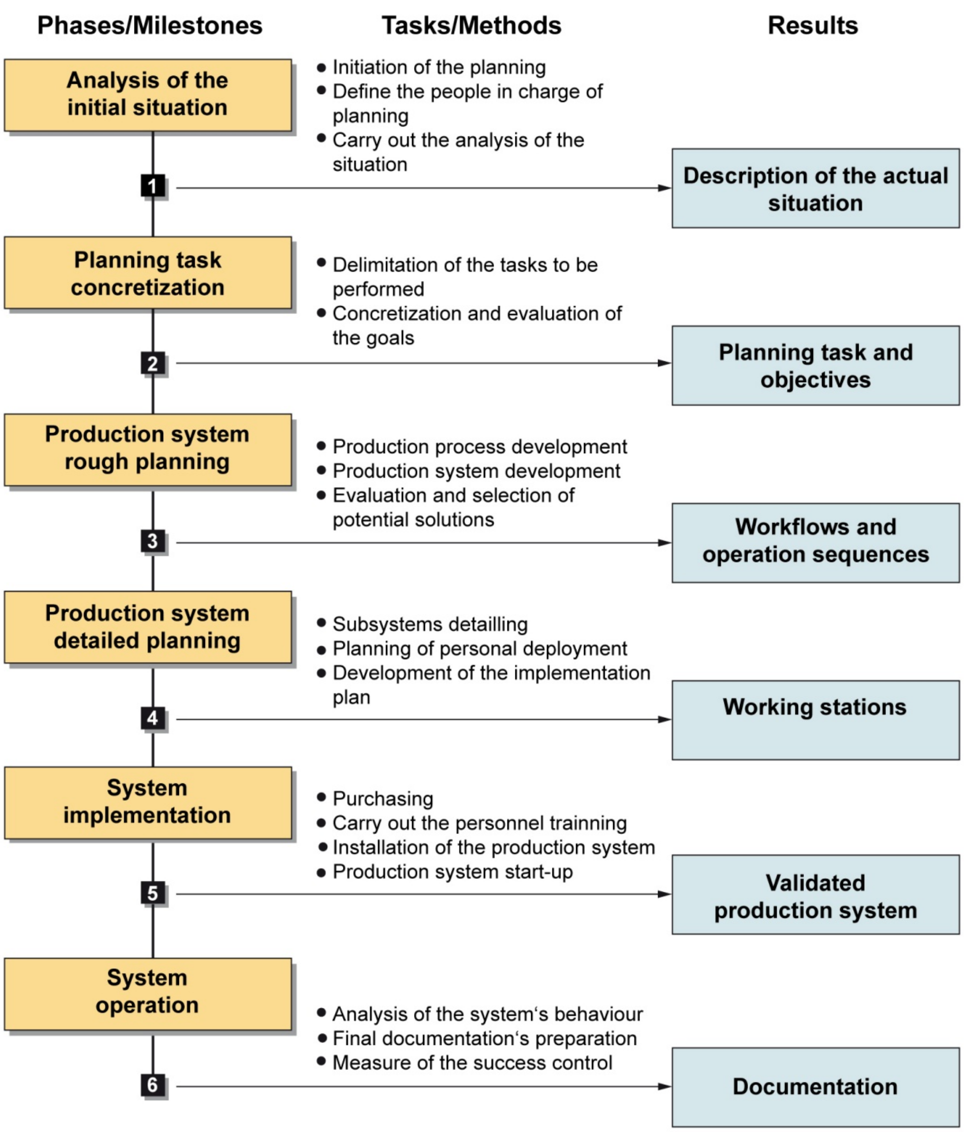

Figure 2-8: Phase model of the systematic for the planning and implementation of complex production systems, [REF87, Pg.89]

A requisite for the systematic is an existent production system. The systematic can be applied at different levels, and it considers not only the technical aspects of a production system, but also the personnel and the organisation. In the following paragraphs, a brief description of the phase model is presented.

Analysis of the initial situation

In this phase, an analysis of the initial situation is made. This analysis includes the task definition, the available resources, personnel, costs, product and production system related information. It also includes determination of the involved areas, to define the development team, the project manager and the decision makers. Based on this analysis, the system's goals are determined [REF87].

Planning task concretization

The planning task to be made, in order to achieve the production system's goals, is defined. Then a definition of the general goal is settled. This goal is divided into organisational, technical, cost and personnel goals. These are further divided into monetarily quantifiable and non-quantifiable. Based on the situation analysis and the rough goals of the planning, the characteristics of the production system are defined.

Production system rough planning

In this phase the production sequence is defined. There are three planning tasks: *Define the production sequence, develop the production system* and *evaluation and selection of possible solutions*. In the first task, the working stations are defined (machining systems), the material plan and finally the information system. In the second task are defined the capacity requirements. Based on them, possible solutions are searched for. In the third task the possible solutions are evaluated and the most suitable solution is selected.

Production system detailed planning

In this phase the subsystems are detailed. The system's specification is developed and the personal deployment is worked out. Finally, an implementation plan is defined and personnel responsible for each task are assigned.

System implementation

In this phase the purchasing of the production system's resources (e.g. machines and handling appliances) starts. The employees are trained and the production system installation is made. Tests are run to prove the system. Based on the tests and experience of the developers and people involved, the system is optimised.

System operation

Once that the test-phase is over, and the system proves its reliability, the system is released. The project-related final documentation is compiled and a success control is performed.

2.2.2.5 Systematic for the process sequence planning and the resources planning

Systematic for the process sequence planning

A milestone for the manufacturing system planning is the process sequence and the resources planning [FSB+08] [REF87] [PBF+07] [Wu92] [Suh01] [Fal00] [Wil08]. There are different methods and approaches for the process sequence definition [REF91a, Pg. 181ff], [Eve97, Pg. 23ff], [Wie97, Pg. 202ff], [War93, Pg. 279]. Nevertheless, according to MICHELS, all this approaches are similar and can be summarized into a six step systematic for the process sequence planning [Mic06, Pg.47]:

- **Analysis of the manufacturing documents:** Manufacturing documents contain all the necessary information required for the manufacturing of the product's shape and surface. In this phase the product requirements demanded to the manufacturing system are analysed and, if necessary, complemented. In some cases, the manufacturing documents contain *explicit information* about the required materials, processes, technologies and resources. Nevertheless, this explicit information narrows down the manufacturing possibilities. Furthermore, in some cases, this information gives preference to a specific field, e.g. the mechanics. As a consequence, the different areas trip over each other, hampering the finding of an overall solution [Mic06].
- **Stipulation of the initial part:** With "initial part" are meant the initial material elements to be used for the product's manufacture. These elements can be raw materials, components from suppliers and trade goods, as well as unfinished and finished goods. For the selection of the initial part, technological, economical and time aspects must be considered [Mic06].
- **Determination of the process sequence**: In most of the cases, for the process sequence determination, an iterative procedure is required. According to KLOCKE and FALLBÖHMER a technology chain is the abstract combination of technologies in a defined order. A critical point for the chain formation is the interaction between technologies [CFF+02]. There are technologies that complement each other, other ones are mutually interchangeable (substitutable) and other ones are mutually exclusive (conflictive). For this reason, combination analysis should be implemented, like the one proposed by BERGER [Ber06]. The process sequence is derived from the technology chain. The initial process sequence describes the value-added process sequence. During the process se-

quence concretization, the handling, quality control and storage processes are included on the description. For the combination of processes and technologies an important point to take into account are the *intermediate states of the product*. The intermediate states are the outputs of the processes and technologies. The outputs present the sequence of the occurred transformations within the product. Via the comparison between the *required* transformations (e.g. product's characteristics) and the *manufactured* transformations, it is possible to evaluate the capability of the processes and technologies. This evaluation makes possible the technologies and processes selection. The *general performance* of the process and technology chain can be validated by comparing the characteristics and functionality of the end product with the stipulated requirements [Mic06].

- **Resources' selection**. Based on the process sequence the resources selection is made. The criteria for the resources selection are the number of items to produce, technical information of the machine, working sequence, and order data. The selection includes tools, machines and appliances. The goal of this phase is to use in the best possible way the available resources but also give an opportunity to new technologies [Mic06].
- **Target time determination.** In this stage the basic information for the capacity planning and scheduling is defined and collected. This information is also used for the calculation of costs and the investment planning [Mic06].
- **Documentation of the work schedules.** In this phase the working plan is documented. This includes the operation planning and the machining operations. The operation sequence is the base for the creation of NC-Programs [Mic06].

Systematic for the resources planning

Resources planning is concerned primarily with the machines and equipment planning. In his work, MICHELS analysed different systematics. Based on these systematics, he summarized a six step systematic for the resources planning [Mic06, Pg.50]:

- **Analysis of the production task:** An analysis is made to determine the required products' characteristics, its related manufacturing process and the processes' frequency of occurrence. This analysis is based on the order data and the manufacturing documentation [Mic06].
- **Creation of products' families**: Products' families are created based on different criteria used for the machines and equipment selection. The criteria selection depends on the type of product and the type of production system [Mic06].
- **Planning of the resource's art and properties**: The products' families are the base for the resources planning. In this phase a *properties profile* is defined for each machine and equipment. This profile is designed to fit with the manufacturing tasks and the products' families.

- **Establishing the capacity requirements and number of machines**: The capacity planning is related to the machine utilization. In this stage the capacity requirement of the family groups and the operations is planned [Mic06].
- **Planning the structural arrangement:** Based on the properties' profile and the capacity requirement, the required machines can be derived. On the basis of the working sequences and the desired production principle (e.g. automated, flexible) the resources are concatenated and a layout is sketched [Mic06].
- **Evaluation and selection**: In the previous steps the capability of the resources was the focus. In this stage, the focus is the economic viability. For this purpose, the collected information from the last stages is used for the resources evaluation and its optimal layout [Mic06].

This systematic offers a good procedure model for the manufacturing system planning. Nevertheless, the steps are not completely defined. The tasks are defined but how these tasks can be made is not explained.

2.2.2.6 A description model for the production system planning

A **model for the production system planning** was developed by FELDMANN et. al.. The model is based on three description-models: Product, process and resource. The *product model* represents the manufacturing task. The *process model* represents the manufacturing processes and the *resources model* describes the available resources for the production [FSB+08].

Essential information for the manufacturing tasks can be derived from the **product information**. Besides shape related characteristics and information about the product's geometry and structure, also technological characteristics like the required tolerance, the surface and the required material can be found. Furthermore, administrative and order related information from the product, influence the planning [FSB+08].

The **process information** contains knowledge about the manufacturing process, as well as *procedural knowledge* for the processes' suitable selection. The procedural knowledge captures the manufacturing system planner's knowledge in form of rules. Some examples of the information here contained are manufacturing specifications and conditions, as well as selection rules for the processes [FSB+08].

Based on the identified processes for the product's manufacturing, suitable resources can be selected. The **resources information** contains facts and data about the resources. Resources enable the formation and modification of the product's form, size, composition, material properties and quality. These resources will constitute the production system [FSB+08].

This model presents an organised structure of the information required within the production system planning. Nevertheless methods are not presented. Table 2-1 presents a summary of the information contained within the three description models.

Table 2-1: Relevant information for the production system planning, according to Feldmann [FSB+08].

Product	Process	Resource
Administrative characteristics **Shape related characteristics** • Product structure (e.g. Bill of materials) • Geometrical data • Form features and semantic • Materials • Single components' and subassemblies' descriptions • Variants **Characteristics related to the order** **Technical characteristics** • Tolerances • Heat treatment condition • Surface quality of finished parts	**Performance** • Description of the process result **Category** • Manufacture • Assembly • Control **Type** • Value-added process • Non value-added process **Alternative process** **Rules for the process selection** **Technical setting parameters**	**Structural unit** • Installation location **Means of fabrication** • Type • Technical setting parameters • Processing time • Unscheduled downtime • Scheduled downtime • Acquisition costs and variable costs **Organisational unit** • Target time • Quality features **Personnel**

2.2.2.7 Manufacturing system's principle solution

Various definitions and models for manufacturing system design and production system design, as well as its contents, appear on the literature. Nevertheless, the essence of these definitions is the same. In the following paragraphs, relevant definitions are presented and common keywords are marked in bold. At the end of the chapter a definition for **manufacturing system's principle solution** will be derived.

DANGELMAIER defines a manufacturing system as a technical and organisational allocation of potential factors for production purposes. A manufacturing system consists of (elementary) working systems. The working systems are the smallest entity that results from the combination of the potential factor **resource** and **labour force**. These entities execute one or more **classes of transformations** [Dan01, Pg.5].

On the other hand, according to SUH a manufacturing system ranges from a manufacturing cell to a large factory that produces a variety of products, services and information. It is a subset of the production system which consists of people, “things” and information. “Things” range from factories to machines, materials, transporters, etc. The design process of AD (see Chapter 2.2.2.3) can be summarized as follows: the **customer attributes or needs** (CAs) should be mapped to **functional requirements** (FRs), which are satisfied by **design parameters** (DPs). DPs are physically implemented by **process** variables (PVs), which are typically quantities such as temperature, pressure, and flow rate that can be changed on **equipment** to yield a desired output value of a product characteristic. In AD decomposition process is used. It starts with the transformation of customer needs to functional requirements (FRs), to design parameters (DPs), and then to process variables (PVs), in so doing the four domains mentioned above are crossed [Suh01].

According to the model of FELDMANN et. al., the required information for **the production planning can be clustered in three domains: product, process and resources**. The model for the production system planning is depicted as three interconnected partial models. The product model represents the manufacturing task. The process model represents the manufacturing knowledge and the resource model describes the means of production for the system [FSB+08].

In accordance to the VDI and REFA the production organisation is constituted by **development**, **purchase**, **manufacturing** and **quality control**. **Manufacturing** comprises manufacture and assembly plus the intern transport, data processing and embodiment of the work systems, planning of the capacity, material, information and manufacturing planning, as well as arrange, monitoring, and ensure the program & orders execution and performance [REFA91].

REFA defines a flexible manufacturing system as a multistep complex production system, conformed by three technical partial-systems: **Machining and processing system**, **assembly system**, **material flow system** and **information system** [REFA87].

NYHUIS et.al. consider a production system as a sociotechnical system. The system has “**inputs**” (for example “Know-how”, methods, materials, financial resources, energy) that are transformed through **value-added processes** (e.g. manufacturing, assembly) and **associated processes** (e.g. transport) into “**outputs**” (e.g. products, remnants). The production system’s task is the fabrication of an intermediate product (subassembly) or a final product. The **processes** are assigned to **resources** (e.g. technical resources, workers). Technical resources are manufacturing and assembly equipment, devices for the warranty of quality, information systems and, in general, the necessary infrastructure required for the execution and maintenance of the production system. The production system is the interaction of organisation, resources, workers and methods [NHR+08].

Based on the above mentioned definitions, the **manufacturing system’s principle solution** can be defined as a first rough description of the manufacturing system. The system presents a collection of interrelated **resources** that perform a set of activities (**processes**) converting goods (**inputs**), according to a set of **requirements** and needs, into a variety of products, services and information (**outputs**) in the most possible cost effective way.

2.3 Problem definition

Customer requirements are becoming more demanding: higher product variety and complexity, shorter product lifecycles and faster delivery times. Companies need continuous product development activities to remain competitive in the marketplace, but, the product is strongly influenced by the production technologies, therefore, product and manufacturing system must be concurrently designed. To accomplish the product and manufacturing system integration, early coordination between the product and the manufacturing system is required.

A possible bridge to overcome this gap is to provide a language for the conceptual design, development, description and documentation of the principle solution of manufacturing systems at an early stage of the product development process. In the following paragraphs, crucial aspects for the development of this language will be treated.

Contents of the manufacturing system's principle solution

To accomplish the integration of the product and its manufacturing system, the manufacturing system's principle solution must be based on the **principle solution of the product** (see Chapter 2.2.1.2) [VDI2206] [GPW09]. Analogue to the product's principle solution, the **manufacturing system's principle solution** needs to be divided into aspects in order to achieve a comprehensive description of the system. These aspects need to be mapped on computer by partial models. These partial models have to be integrated into a coherent system of partial models.

There is no general accepted procedure to design manufacturing systems [Lin01] [SJ08]. Nevertheless, some common concepts and tasks can be derived from the approaches mentioned in Chapter 2.2.2:

- **The manufacturing system design cannot be seen as a stringent sequence of process steps.** It is more an interplay of tasks within an iterative process. Each developed task gives feedback to other tasks and influences the complete design. For this reason an adequate exchange of information is crucially important.
- **The manufacturing design process starts at an early stage of the product development process, indeed, with the product requirements definition**. These requirements include the attributes to manufacture, materials, the shape, structure of the product (building structure) and the desired production rate.

- **The product is strongly influenced by the production technologies.** Therefore, the manufacturing system design and the product must be concurrently designed, with an integrative approach.
- **To assure the product and manufacturing system integration, an holistic view of the system must be implemented.** To assure a good product and manufacturing system conceptual design, the product and the manufacturing system must be considered as whole system rather than simply its individual components.
- **Suitable process must be selected for the attributes' manufacture and later, suitable process sequences must be determined.** The assembly process, among other things, depends on the product's structure. The material and the required production rate influence the process selection. The processes selection has influence on the performance of the system and its cost, as well as on the quality.
- **Resources must be assigned to each process**. The selection and assignment depends, amongst other things, on technological criteria like the availability of the resource, the tolerances, etc. Additionally, the required product properties, and the price of the product affect the resources selection.

Summarizing, a **manufacturing system's principle solution** (see Chapter 2.2.2.7) presents a collection of interrelated resources that perform a set of activities (processes), according to a set of requirements and needs, converting goods (inputs) into a variety of products, services and information (outputs) in the most possible cost effective way. Therefore, the principle solution needs to cover and **represent** the necessities of four **domains**. These domains are the **requirements** to be fulfilled by the system, the **processes** to be made in order to fulfil these requirements, the **resources** to be applied within the system, and the **structure** or layout of the system (**shape**), as well as their **interconnections**.

The conceptual design process, a continuous and iterative process

For modelling the manufacturing system it is required an iterative process, which starts with an abstract concept. Each iteration loop within the development process concretizes the system. The system's development consists of many process steps. Some of these steps must be done in parallel. Due to the strong relations and dependencies between the elements of the system, there are many interfaces between the process steps. Systems are not static. They must be designed and improved over time until the system is phased out. For these reasons, the specification must be able to represent the system's concept with different levels of detail and abstraction. *Decompositions, aggregations* and *extensions* of the specification must be considered.

Summarizing, for modelling the manufacturing system it is required an **iterative process**, which starts with an abstract concept. **The concretization, changes or additions**

on the system's information must be possible. The goal is to specify the principle solution during the system's development. The specification must be an aid for the conceptual design, development, description and documentation of the principle solution of a manufacturing system.

A common language for the conceptual design

To accomplish the integration of the product and its manufacturing system, early coordination between the principle solution of the product and the manufacturing system is required. Nevertheless, the early coordination between the principle solutions requires a **unified language** that supports the **communication** between the interdisciplinary design team and coordination between the design activities across the entire design process. This is especially important during the conceptual design stage of complex products. Due to the product's complexity, the development results must be specified successively with the development process. The concretization of the design must be described and documented, not only to enable an integrative design but also to support constructive discussions within the interdisciplinary development team.

This implies that a holistic view must be afforded to facilitate an integrative design. As already mentioned, an inherent difficulty with this approach is the fact that the task involves experts from different disciplines. Each discipline has its own vocabulary and working methods. This complicates even more the exchange of information.

From an educational and an industrial perspective, providing manufacturing-specific concepts and their integration with product modelling concepts are a matter of special interest. Specialized languages provide suitable terminology for the domain of interest. Therefore, the modeller does not have to reconstruct generic terms, into more specific ones, but rather reuse the already proved domain terms [Fra99] [SJ08]. This fact increases the comprehensibility and clarity of the models. Furthermore, the use of specialised concepts improves the chances to check a model's integrity on a syntactic level [Fra99].

Models depicted with a graphical representation contain information in a clear and understandable form. A graphical description within a modelling language is of special interest because they allow the visualization of complex relationships within a system [HH91].

A type of graphical modelling language are **semi-formal specification techniques** (see Chapter 2.1). A semi-formal specification is used for encoding and decoding information. It has graphic constructs and modelling rules. With a graphic notation the models are relatively easy to understand and are suitable as communication medium. Visualization brings a deep understanding and leads to major breakthroughs in the modelling performance. Furthermore, a graphic notation leads not only to a better understanding but also provides a common language between the involved parties. This facilitates a consensus that enables and enhances implementation.

In conclusion, for the manufacturing system design a specialized graphical semi-formal specification technique is needed, because they provide the suitable terminology, graphics and modelling rules for the design.

A domain-spanning model for a complex system

Manufacturing systems are complex systems, affected by dynamic markets and a diversity of actors. JOSLYN and ROCH define a complex system as following:

> *„A* ***complex system*** *is commonly understood as any system consisting of a large number of interacting components (agents, processes, etc.) whose aggregate activity is non-linear (not derivable from the summations of the activity of individual components), and typically exhibits hierarchical self-organisation under selective pressures“ [JR00, Pg. 71].*

Thus, the manufacturing system's development environment is an unstable environment, characterised by frequent changes in design criteria and by huge pressure to compress project delivery times. Communication and negotiation are the base of the development process. Furthermore, interdisciplinary development teams are involved and practitioners must search for innovative ways to structure the design process.

Involving different methods for taming the complexity is one way to overcome these problems. There are different methods, like the *modelling paradigm of different views of a system* [NAB04], *different levels (or modules) of organisation* [Agr03] [VR08], *structuring concepts* (e.g. according the functionality) *and hierarchies* (e.g. according the assembly sequence). The specification should apply at least two of the mentioned methods in order to assure a smooth knowledge transfer between the involved disciplines and processes.

The *modelling paradigm of different views of a system* is a type of framework that supports an abstract description of *socio-technical systems*[9] and its running processes. The concept is characterised by the description of complex system situations by using different projections, also named views. Through this model construction isolated abstractions of the systems are created [NAB04].

Modelling complex systems from the point of view of *different levels (or modules) of organisation* means to analyze the system and search for multiple parts "nested" inside each other. In this type of models, the systems are represented by levels. With "nested" is meant that parts located inside these levels interact among each other stronger than between the levels. This perspective from the system gives a certain degree of abstrac-

[9] "*The concept of the socio-technical system was established to stress the reciprocal interrelationship between humans and machines and to foster the program of shaping both the technical and the social conditions of work, in such a way that efficiency and humanity would not contradict each other any longer." [Rop99-ol]*

tion. In this way it is possible to concentrate on the description of mechanisms of just one or two neighbouring levels [Agr03] [VR08].

Summarizing, a manufacturing system is a complex system. For taming its complexity a domain-spanning model of the system is required. This model must represent the system as an harmonic composition of interrelated views, with different levels of organisation and abstraction degrees.

2.4 Requirements

The goal of this work is the development of a specification technique for the conceptual design of a manufacturing system based on the principle solution of the product. A success factor for this task, is the definition of a manufacturing system model and data-model. Based on the problem definition stated on Chapter 2.3, the following requirements were derived:

Requirement R1 - Holistic description of the manufacturing system's principle solution

It must be able to represent the elements, activities, relationships and interactions that occur within a manufacturing system. The specification must describe the system as a whole and not only the elements from which it is comprised. For that purpose, the requirements on the system, its basic logical and physical operation, its structure and the essential characteristics of its elements must be specified. The specification must include the shape and behaviour of the system as well as the relationships and interdependences between the system elements and their characteristics.

Requirement R2 - Description of a manufacturing system at an early stage of the product development process

The specification must take into account the relationships and interdependences between the product and the manufacturing system conceptual design. The specification must be able to depict a first concept of a manufacturing system, based on information taken from the product's principle solution.

Requirement R3 - Consistent semi-formal specification technique

A semi-formal specification technique facilitates the comprehensibility of the model and makes it suitable as a communication medium. The specification must be consistent, i.e. free from variations and contradictions, showing a steady conformity to vocabulary and methodologies among the involved disciplines, aiding the team work in an interdisciplinary environment.

Requirement R4 - Domain-spanning specification technique

The specification must depict and document a domain-spanning visualization of the manufacturing system's principle solution, allowing later, in the *embodiment design* *phase*[10], to concretize it in specific domains. For this reason, a balanced consideration between the involved domains (*requirements, process, resource* and *shape*) is required.

Requirement R5 - Support the iterative process of the conceptual design's development and its concretization

With *support* it is meant that the specification should assist the developer during the conceptual design's typical iterative process and also with its concretization. The specification must be able to describe and document the manufacturing system's concept from its starting point, the principle solution of the product, until its final state, the manufacturing system's principle solution. In other words, it must represent the system with different levels of detail as well as different levels of abstraction.

Requirement R6 - Reduction of complexity

To make the specification more understandable is convenient to have a specification fractioned in suitable parts and concepts and assort these in a worthy structure, taking care that the mentioned structure is still manageable. Furthermore, it is recommended to recourse to diagrams and views in order to simplify and clarify the model. With complexity reduction is not only the structure meant, but also the semiotic[11] of the language, because the language must be suitable and comprehensible for people of different disciplines.

Requirement R7 - Expandability

There are many types of complex systems (mechatronic systems, nano systems, etc.) which demand specific requirements from the manufacturing system. To fulfil specific requirements of each system's concept, it is necessary a specification that allows the fill in of special requirements and additional constructs. For these reasons, it is needed a „basic specification", or a core specification technique with **indirect integration.** In other words, if required, the user of this specification can extend the specification (e.g. add a view, add constructs and relationships, etc.) in order to depict specific requirements that further concretizations of the principle solution may need. These extensions may be aggregated within separated modules.

Requirement R8 - Communication medium

The specification technique must provide a suitable communication medium, because the early coordination between the principle solutions requires a unified language that supports the exchange of information between the interdisciplinary design team and

[10] "*The Product Development Process has four phases: Task Clarification, Conceptual Design, Embodiment Design, and Detail Design. (...) During the embodiment design phase, designers start with the selected principle solution and work to produce a definitive layout of the proposed technical product or system in accordance with technical and economic requirements." [PBF+07, Pg.xiii]*

[11] "***Semiotic** is the study of signs and symbols and their use or interpretation." [Oxf10-ol]*

coordination between the design activities across the entire design process. The challenge is to generate a visualization that engenders a common understanding of the principle solution between the involved parties.

3 State of the Art

3.1 Specification techniques

In the following section, a review of existing frameworks, methodologies and specifications for the manufacturing system design is made. Each approach is described and then evaluated with the requirements stated on Chapter 2.4. At the end of the chapter, a summary of the evaluation is presented. Additionally the need for action is discussed.

3.1.1 GRAPES

GRAPES is a graphical modelling language for formally describing a system. **GRAPES has a formal, defined text and graphic syntax.** It uses an object-oriented model structure. This means that information processing systems are viewed as systems comprising objects which communicate with each other. GRAPES is used for modelling information processing systems, nevertheless, it can be used in organisation engineering, systems engineering and software engineering. Aspects of the system development like project management, documentation system that are not covered by GRAPES, are completed by DOMINO. DOMINO is an integrated process engineering technique for the development and the domination of complex information systems [HH91].

The development of GRAPES (Graphical Engineering System) arose from the requirements of the System Planning, Application Software and Projects department of Siemens AG. GRAPES includes basic elements of the *Input/Output requirements language* (IORL), *Structured Analysis* (SA), *Structured Analysis and Design Technique* (SADT) and *Specification and Description Language* (SDL) [HH91].

GRAPES permits object-oriented modelling, abstraction for data types and process as well as data modelling using entity-relationship diagrams. It includes a dynamic processing model, supports concurrency, synchronous and asynchronous communication concepts [HH91].

The models in GRAPES are represented by means of diagrams, symbols (see Figure 3-1) and tables. The diagrams allow the structure, process and information of a complete system to be represented in the form of models and to be gradually refined. Each construct in the graphical representation of GRAPES correspond to one construct in the GRAPES textual language. The relationships between the graphical symbols, diagrams and tables also have their counterparts in the textual language [HH91].

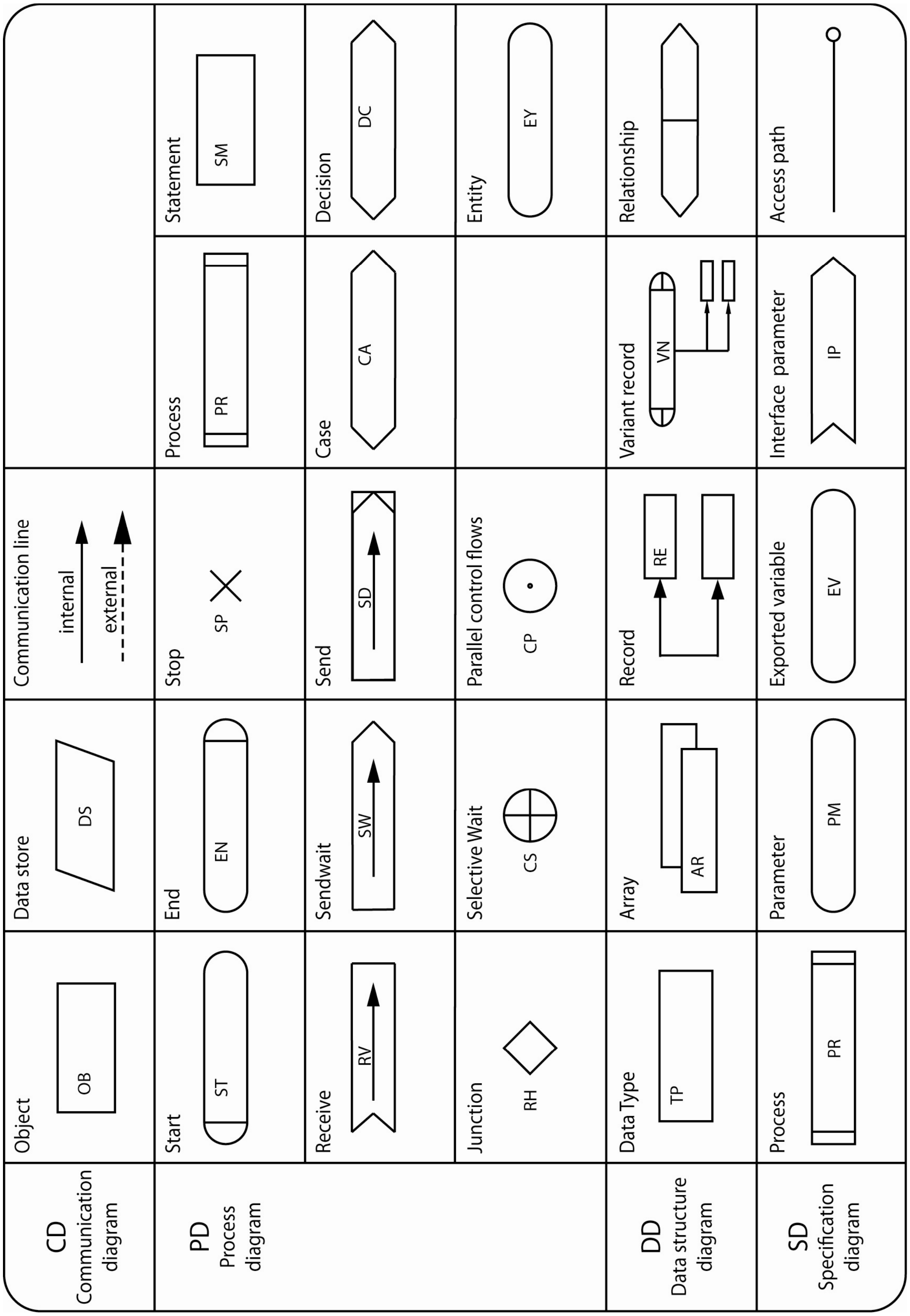

Figure 3-1: Summary of GRAPES symbols, [HH91]

The available diagrams types are described briefly below [HH91].

- **Diagrams for modelling structures and communications**
    - Communication Diagrams (CD): Used to describe the way in which an object is made up of sub-objects and to represent the communication relationships between the sub-objects (communication lines)
    - Interface Table (IT): Assigned to each communication line in a communication diagram. Describes the structure of the communication lines in terms of channels and associated data types
- **Diagrams for modelling behaviour and processes**
    - Process Diagrams (PD): Used to define the behaviour of process objects (process), procedures and functions
    - Data Tables (DT): Are used for declaring constants and variables to which processes, procedures, functions and modules are applied
    - Specification Diagrams (SD): It is used for describing interfaces. They represent the call interfaces (parameters) or procedures or functions and the export interfaces of modules
- **Diagrams for modelling data**
    - Data Diagrams (DD): Allow data structures to be modelled. The relationships between sets of data can be described here, as user defined data types
- **Diagrams for representing the declaration hierarchy**
    - Hierarchy Diagrams (HD): These are used to represent the interrelationships between the documents defined in the model.

Figure 3-2 presents an example of a process diagram, named *store_administrator function*. The diagram explains the functions of a store administrator. The store administrator executes two possible processes (*process* and *add_to_stock*), depending on the receiving orders (*in_writing issue, outgoing_goods issue* or *incoming_goods addition*). For example, in case of *incoming_goods*, the administrator adds the goods to the stock (process *add_to_stock*). In case of *outgoing_goods*, the administrator *dispatches* the goods. In case of *shortage*, he or she *replenishes* the stock and in case of *out_of_stock* he or she *reorders* goods.

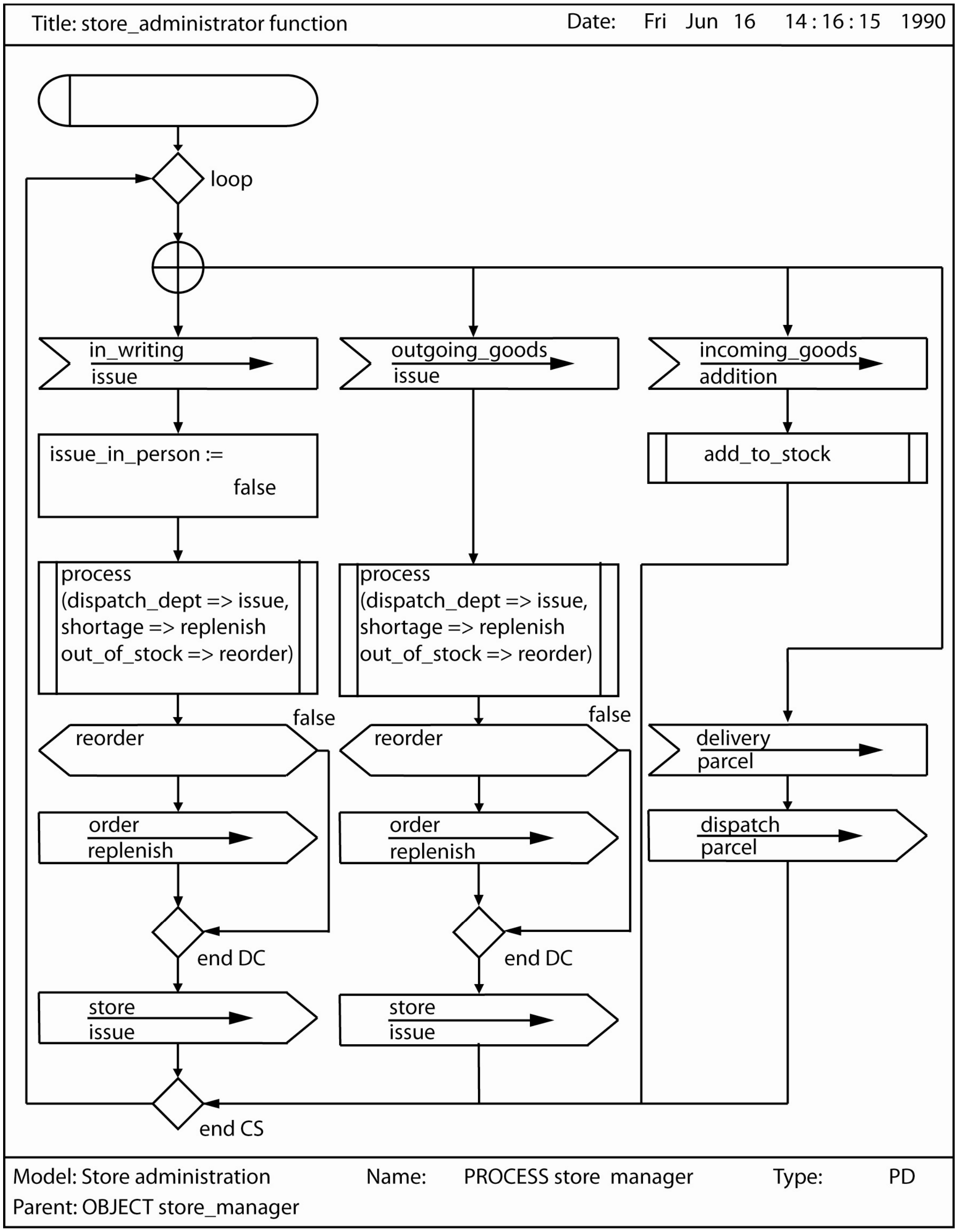

Figure 3-2: Example of a process diagram [HH91, Pg.131]

GRAPES Evaluation

GRAPES arose from the necessity to synthesize data processing, office automation and communications towards an integrated information and data processing technology. For these reasons it counts with typical terminology used for software and information systems. It uses generic terms, such as “object” and “relationship”. By reconstructing these generic terms and with the implementation of constraints it is possible to model a manufacturing system, nevertheless is difficult to model the resources' and system's shape. Furthermore, depending on the reconstruction of the language and constraints, it could be possible a domain-spanning description of the system.

GRAPES is a graphical and formal modelling language. It supports the iterative process of the conception's development, and reduces the complexity of the system with the aid of diagrams and hierarchies. Nevertheless, the focus resides in the design of information processing systems, rather than in the manufacturing system design.

Due to the GRAPES' information system background, it counts with communication concepts (e.g. dynamic processing model, concurrency, synchronous and asynchronous communication, etc.) that aren't necessary at an early production system's conceptual stage. Furthermore, due to its dual nature (graphical and formal) the language has a complex syntax and grammar. Each construct in the graphical representation of GRAPES corresponds to one construct in the textual language. It was not designed to be used as a communication medium. No extensions (expandability) are allowed.

3.1.2 Integrative specification of the product's and production system's conceptions

MICHELS created a semi-formal specification technique for the parallel and integrative development of product and production system. It is based on the "Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme", by KALLMEYER [Kal98] as well as on the "Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme" by FRANK [Fra05].

The aim of this specification is a first description of the product's structure and functionality as well as its production system. With this purpose in mind, the product and the production system are seen as the composition of different views. The product is described with: *Requirements, Functions, Active structure, Behaviour, Shape and Intermediate State*. In the other hand, the production system is described with *Process sequence, Production structure, Behaviour* and *Shape.* These views are mapped on computer by partial models (see Figure 3-3) and were made for the development progress. These document the conception's process and contain models, data and relevant information required within the design process.

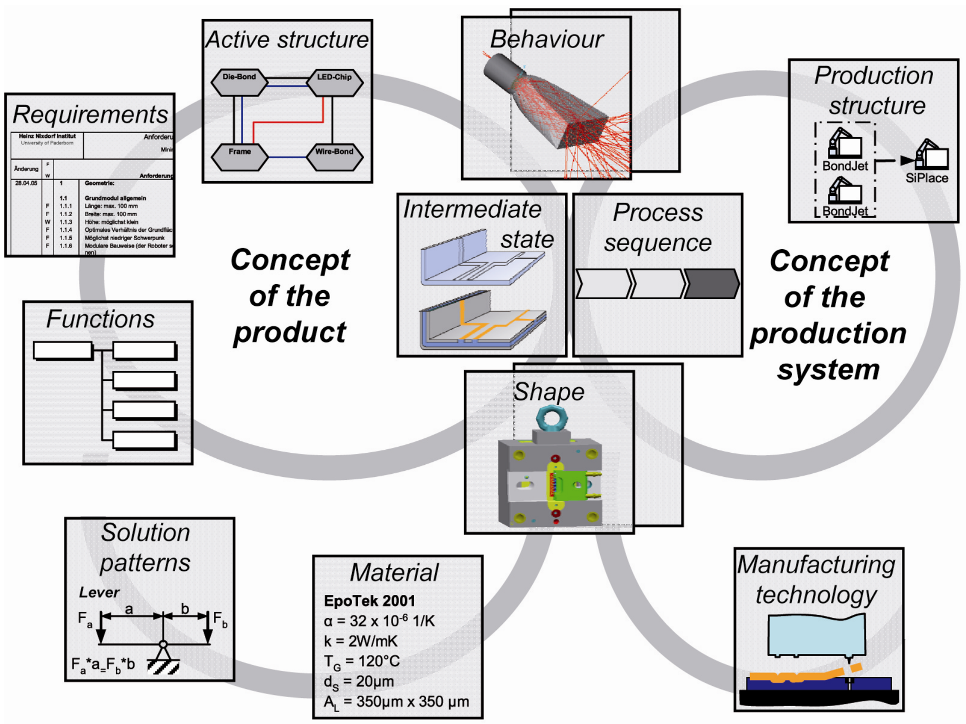

Figure 3-3: Partial models for the integrative specification of the product's and production system's conceptions, adapted from [Mic06, Pg.90]

The partial's models of *Behaviour* and *Shape* are double because these models are in both conceptions. The partial models *Solution Patterns, Material* and *Manufacturing technologies* are models that complement the other models, contain already known information and bases for the design, due to this, they play a secondary role. These partial models are organised below the principal models [Mic06].

In Chapter 2.2.1.2 a brief description of the product partial models was presented. In the following paragraphs a brief description of the remaining partial models and their elements will be given.

Process sequence: This partial model describes the logical sequence of the production steps. There are two types of production processes: manufacturing process and assembly process. Individual manufacturing processes are used to show the transformation of a raw material or component, step-by-step into final product. In each manufacturing step, the component acquires the attributes specified within the requirements list. In the other hand, assembly processes are used to show the joining of parts or subassemblies until the final assembly is achieved.

Intermediate state: It represents the sequence of changes (intermediate state) that the parts and subassemblies experience within the production system. The transition of one intermediate state to another is executed by production processes.

Production structure: It comprises the required production equipment and relevant information for the production. Logical relationships are considered. The material flow between the resources is depicted nevertheless, it is not closer examined.

Shape: There are two partial models *shape*, one for the product (see Chapter 2.2.1.2) and one for the production system. The partial model of the production system deals with the resources' geometry, e.g. the permissible floor space area.

Behaviour: There are two partial models *behaviour*, one "set" for the product (please refer to Chapter 2.2.1.2) and one for the production system. The latter one describes the interaction between intermediate states and production processes during production. The *behaviour* results from the interaction of the manufacturing technology and the machine parameters, e.g. the injection pressure and the tool temperature during the injection moulding [Mic06].

Solution Patterns: "Solution pattern" is used here as a general term. A solution pattern describes the solution's core of a reoccurring problem [GFD+08]. This partial model represents the elements of a possible solution.

Material: In this partial model are specified relevant material's properties and characteristics for the product, subassemblies and parts.

Manufacturing technologies: This partial model is used for the production processes. Here are represented the manufacturing function and the relevant properties and characteristics of the selected technologies.

The partial models (PM) are interconnected. These models and their cross-linking are modelled by means of a graphic notation, which consists of constructs and modelling rules. Figure 3-4 shows how the partial models active structure, intermediate state, process sequence and production structure are connected. Intermediate states, in PM intermediate state, and system elements, in PM active structure, are linked with sequence relationships. Production processes, in PM process sequence, are also linked with sequence relationships. Manufacturing resources, in PM production structure, are linked with material flows. Finally, production processes are linked to intermediate states, system elements and to manufacturing resources with realisation relationships.

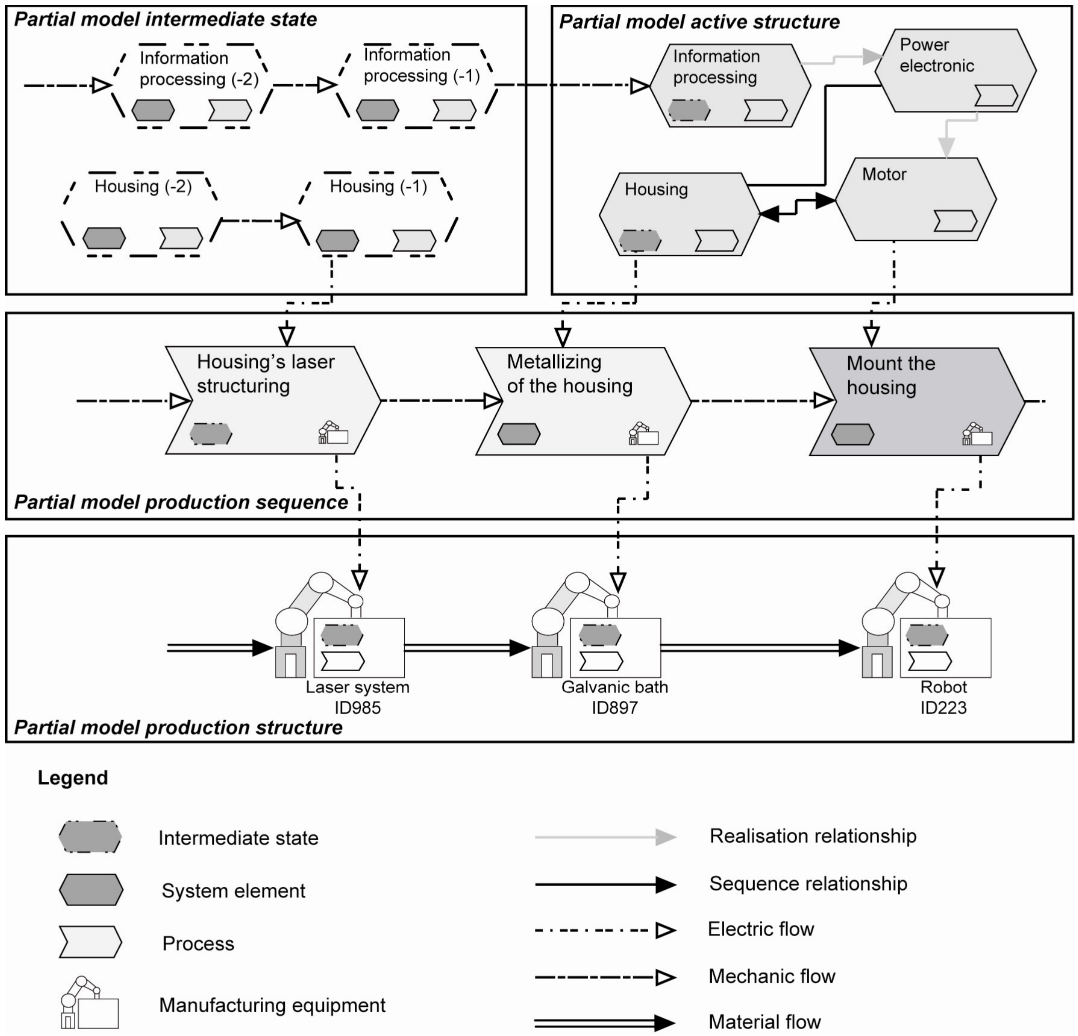

Figure 3-4: Connections between the partial models active structure, intermediate state, process sequence and production structure, [Mic06, Pg.121]

Evaluation of the integrative specification of the product's and production system's conceptions

This is a semi-formal specification designed for the product's and production's system conceptions. It is possible to make a rough description of the production system at an early stage. It is a domain-spanning specification technique that reduces the complexity of the system by using hierarchies and aggregation relationships between the different elements of the system. If required, it is possible to integrate views or constructs to the specification.

Nevertheless, the focus resides in the product and production system design. For this reason, it counts with concepts that aren't necessary for the production system's design, e.g. a software component's development. In addition, for the production system's description only added-value processes are considered, that leaves the transport and handling of the product (logistics) unconsidered [Mic06, Pg.94]. Furthermore, the product and production system are depicted by a coherent system of integrated partial models. Nevertheless, the integration of the partial models and the data model of the specification are not covered [Mic06, Pg.107]. It was designed to be used as a communication medium. Extensions are allowed.

3.1.3 SysML

SysML is a language created for modelling systems engineering. It is methodology-independent and reuses a subset of UML 2.0[12] (Unified Modelling language) concepts, diagrams and augments with some new diagrams and constructs appropriate for systems modelling. It supports the specification, analysis, design, verification and validation of complex systems.

> *„The Systems Modelling Language (SysML) is a general-purpose graphical modelling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In particular, the language provides graphical representations with a semantic foundation for modelling system requirements, behaviour, structure, and parametrics, which is used to integrate with other engineering analysis models" [OMG09-ol].*

SysML brings two main enhancements into the UML language:

- **Requirement Management**: SysML provides systematic tools to manage requirements, while UML does not have any specialized tool to deal with them. Furthermore, requirements can be allocated to blocks or functions defined within the system model. This enables a trace to identify which parts of the system satisfy the requirements [Hau06].
- **Parametric Diagram:** This diagram represents internal constraints on the system through constraint blocks which bind constraint parameters and properties of classes defined in the constraint context. Constraints can be mathematical equations with parameters bound by block properties. In this way, a broad range of engineering equations can be integrated into a simulation model [Hau06].

With these two main enhancements SysML is intended to support modelling of a broad range of systems, furthermore, SysML is methodology and tool independent. It is a modelling language that provides semantics (meaning) and a notation.

---

[12] UML 2.0 is a language used to specify, visualize, and document models of software systems, including their structure and design. See [OMG09a-ol] for further details.

SysML has four pillars: system requirements, behaviour, structure and parametric relationships. These pillars are specified by means of different diagrams (Figure 3-5). These diagrams can be classified into requirement diagram, behaviour diagrams, structure diagrams and parametric diagram [Hau06].

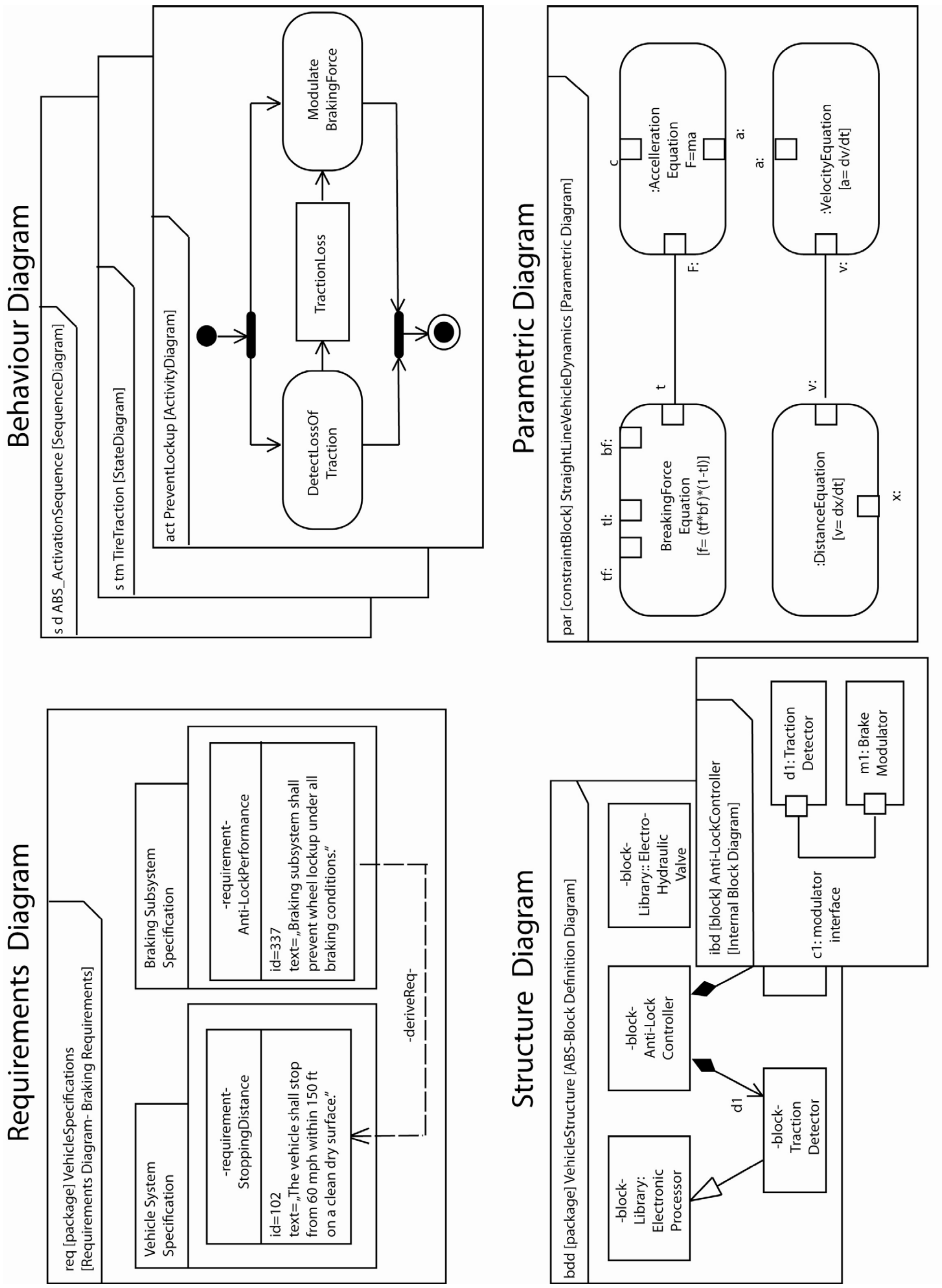

Figure 3-5: The diagrams of SysML for specifying the system requirements, behaviour, structure and parametric relationships [Hau06]

In the following paragraphs a brief description of the pillars and their diagrams will be given.

Structure: Blocks are used as basic structural elements. They provide a unifying concept to describe the structure of an element or system, they have properties and are used to specify hierarchies and interconnections. The blocks are the base of the structure's diagrams. The **structure** has 3 diagrams. The **block definition diagram** (bdd) is used to describe the system hierarchy and system/component classifications. The **internal block diagram** (ibd) is used to describe the internal structure of a system (parts, ports, and connectors). The **package diagram** is used to organise the model by system hierarchy, by domain, etc. [Hau06] [FMS06].

Behaviour: The system behaviour has 4 diagrams, the **use-case diagram**, **activity diagram**, **sequence diagram** and **state machine diagram**. The **use case diagram** provides a description of the system functionality in terms of usages and goals of the system by actors. The **activity diagram** represents the flow of data and control between activities. Activities are used to specify the flow of inputs and outputs and control, including sequence and conditions for coordinating activities. The **sequence diagram** represents the interaction between parts of a system, nevertheless SysML does not include timing, interaction overview and communications diagram. The state machine diagram describes the state transitions and response-actions of the system. It supports event-based behaviour [Hau06] [FMS06] [OMG07].

Requirements: It has one diagram, the **requirement diagram** (req) which captures requirements hierarchies and the derivation, satisfaction, verification and refinement relationships. Requirement relationships model the content of a specification. The relationships provide the capability to relate requirements to one another and to relate requirements to system design models and test cases. The requirement diagram provides a bridge between typical requirements management tools and the system models [Hau06] [FMS06] [OMG07].

Parametric: It has one diagram, the **parametric diagram** (par)**.** It is used to express constraints (equations) between value properties such as performance, reliability and mass properties to support engineering analysis. The constraint block captures equations. The parametrics enable the integration of engineering analysis with design models. SysML includes an allocation relationship to represent various types of allocation, e.g. the allocation of functions to components [Hau06] [FMS06] [OMG07].

Evaluation of SysML

SysML is a visual modelling language that provides semantics and their notations for systems engineering. SysML is intended to support the modelling of a broad range of systems. For this reason, SysML is methodology and tool independent. As GRAPES, SysML uses generic terms, such as "block". By reconstructing these generic terms and with the implementation of constraints it is possible to model an existing manufacturing system. Furthermore, if the necessary frameworks are provided, is also possible to simulate the model [HRM07] [KM07]. Nevertheless, SysML doesn't support the conception process and its iterative development; it was designed for the description. Furthermore for the conception of the system is of prime importance the modelling of the system elements and their interconnections at an early stage of the product design. For the representation of technical connections, (e.g. interdependencies) are necessary several diagrams (e.g. bdd, ibd, par). The model should present an intuitive, appropriate and understandable description. Nevertheless, the complexity of the language, and the usage of large number of diagrams can very easily make the user to loose the overall view. It was not designed to be used as a communication medium. Extensions are not allowed.

3.1.4 Manufacturing process model

According to DANGELMAIER, to describe accurately a manufacturing process two dimensions are needed: The manufacturing sequence dimension and the time dimension. The **manufacturing sequence dimension** is depicted with the **manufacturing sequence graph** and the **time dimension** is depicted with the **time model.** With the aid of these models, it is possible to create a set of diverse manufacturing processes, a solution set, for the manufacturing planning. Through the definition of some restrictions, the set is reduced to an acceptable and as good as possible, manufacturing process [Dan01].

The **manufacturing sequence graph** represents, qualitatively, the possible manufacturing processes and the different states of the involved manufacturing system's elements. The graph structures the manufacturing process according to a decision's need.

> *„A decision's need arrives after a manufacturing process step completion, when the production factors'[13] flow must sum up or spread" [Dan01, Pg. 14].*

Model elements are used to represent the manufacturing process steps and the production factors. Table 3-1 presents an overview of these elements.

[13] Production factors are all the manufacturing system's elements that participate on the manufacturing process' steps [Dan01].

Table 3-1: Model elements for the representation of manufacturing sequences, according to Dangelmaier [Dan01, Pg. 15].

Model's element	Description
Manufacturing Element (M-Element)	• Represent a manufacturing system's element in a determined state. • Description through identification and characteristics.
Manufacturing Element – Category (ME-Category)	• Collects a set of M-Elements through common decision's needs
Manufacturing Element – Node (ME-Node) Symbol: △	• Administrates a hierarchy of ME-Categories
Manufacturing Process Event (M-PEvent)	• Represents a part of the manufacturing process • Description through (Identification) Input and Output objects
Manufacturing Process Event – Category (MPE-Category)	• Collects a set of M-PEvents through common decision's needs
Manufacturing Process Event – Node (MPE-Node) Symbol: ▭	• Administrates a hierarchy of MPE-Categories
Edge Symbol: →	• Connects ME-Nodes and MPE-Nodes • Characterised through ME-Nodes and MPE-Nodes

The **manufacturing sequence graph** presents a set of linked nodes. The nodes are linked through edges. The edges only pass through nodes of different type (ME-Node, MPE-Node). Figure 3-6 presents a manufacturing sequence graph example.

The main process between the initial material (raw material) and the end product, the value-added process, is presented in the graph's middle. The main process is assisted through secondary processes (palletes transport, tool cycle). M-PEvents that represent processing steps, are MPE-Nodes with a "B" assigned. Transport processes are MPE-Nodes with a "T" assigned and commissioning processes are MPE-Nodes with a "K" [Dan01].

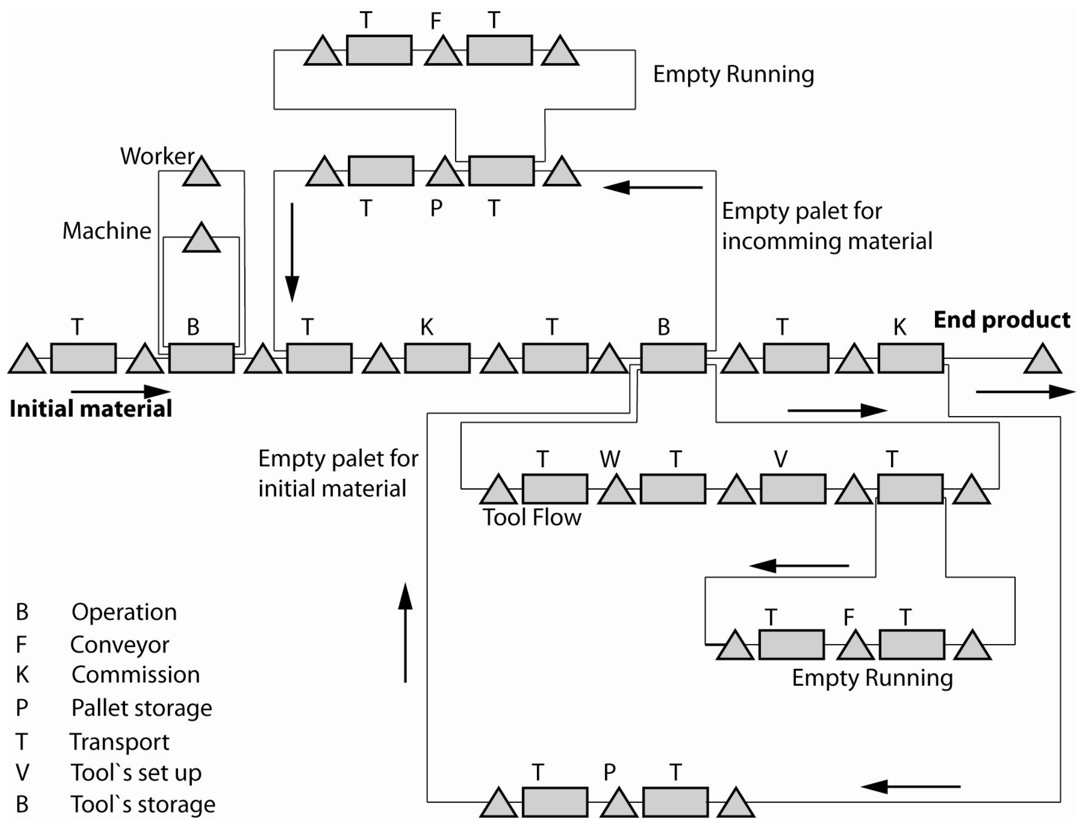

Figure 3-6: Manufacturing sequence graph, adapted from [Dan01, Pg. 23]

The **time model** (Calendar) depicts the time structure. It allows statements about determined manufacturing process' events, specifically about manufacturing system's states. The elements of the time model represent real periods of time or real points in time. A time model example is a complete hours model (24 hours per day, 7 days per week). Another example is the business calendar (8 hours per day, 5 days per week). Only through the definition of the applied time model and its conjunction with the manufacturing sequence graph it is possible to describe the manufacturing system's states and events [Dan01].

Model events represent manufacturing system's real and supposed (past and future) events and states. These consist of 3 descriptions: **factual** (e.g. 50 desks)**, temporal** (e.g. 17.11.07) and **type of event** (e.g. Stock). Model events are depicted with dots [Dan01].

Type of events summarize similar model events. These are modelled as dots. The dots are allocated to the nodes. ME-Nodes have three points (entry, centre, exit) MPE-Nodes have five points (entry, starting M-PEvent, running M-PEvent, ceasing M-PEvent, exit) [Dan01].

The **manufacturing process model** is the result of the combination of the **manufacturing sequence graph**, the **time model** and the **model events.** The level of detail of the manufacturing process model depends on the degree of abstraction and completeness of the manufacturing planning's state [Dan01].

Evaluation of the manufacturing process model

The manufacturing process model offers the possibility to represent manufacturing process within the manufacturing planning scope. It provides constructs that represent manufacturing process' elements and the necessary information for the planning. The user is the one that determines the model's level of detail; the manufacturing sequences can be abstract (many processes aggregated into one single process) or a "one to one" (1:1) model. Additionally, "real" and "possible" processes can be modelled. Furthermore, this model offers the possibility of a computer-assisted manufacturing planning. Nevertheless this model was not designed as a communication medium but rather for the analysis, specification and representation of existing manufacturing sequences and its events for its further employment in planning and optimisation processes. The model does not offer a domain-spanning specification technique. It is focused on the processes and logistics of a manufacturing system. The graphical notation is extreme simple and highly abstract, which makes it counterintuitive. There is no graphical differentiation between the different manufacturing elements (e.g. information, material or resources) and the different manufacturing process events (e.g. manufacturing process, assembly process, handling operation, transport) which is an important aid at the conceptual stage. It was not designed to be used as a communication medium. Extensions (expandability) are not allowed.

3.1.5 REFA's symbology to depict flexible manufacturing systems

REFA -Verband für Arbeitsgestaltung, Betriebsorganisation und Unternehmensentwicklung - is a german organisation that deals with work design, companies' organisation and development. This organisation created a systematic for the planning and implementation of complex production systems. This systematic includes a symbology used for the system's description. According to REFA, a flexible manufacturing system can be divided in 3 subsystems, independent of the system's size or its degree of automation [REF87]:

- **Machining/processing system and assembly system**: Production equipment, machines, tools, devices, inspection devices, auxiliary products, lubricants, etc.
- **Material flow system:** Conveyor resources, vehicles, hoisting devices, conveyor with continuous motion, handling devices, conveyor aid resources, pallet, pick-up elements, grippers, etc.
- **Information's system:** Hardware, computers, terminals, cables, circuits, software, programmes, etc.

Inside of these subsystems persons, technique, organisation and information can be found. To represent the three subsystems, five symbols are used: material flow, information flow, material flow system, information system and machining, processing, and assembly system. These symbols allow the representation of the interactions of the system, nevertheless, more specific information of the system is not considered. Figure 3-7 presents the interaction between the three systems within a flexible manufacturing system.

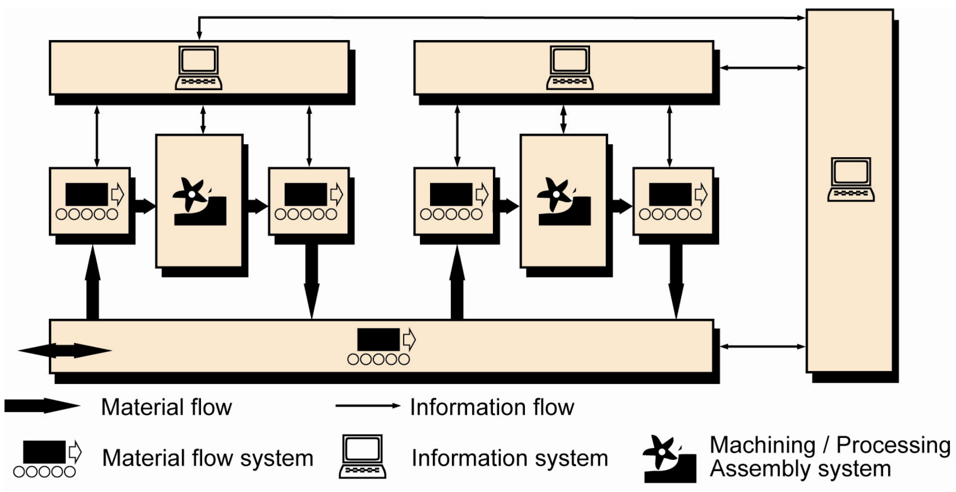

Figure 3-7: Schematic representation of a flexible manufacturing system [REF87, Pg. 50]

Evaluation of REFA's symbology to depict complex production systems

With the here presented symbology it's possible to represent a production system in three dimensions: machines, material flow and information. This approach present's an easy way to illustrate the interactions of the system, nevertheless the representation is really abstract. It does not facilitate a suitable description of the system, because there is no representation of the system's elements but rather a representation of the manufacturing system's subsystems. Furthermore, the focus is on the description rather than in the conceptual design. The model does not offer a domain-spanning specification technique. An expansion of the symbols is not allowed. It was not designed to be used as a communication medium. Extensions (expandability) are not allowed.

3.1.6 Process mapping

FRANK GILBRETH developed the first process mapping system, called "Process charting". This mapping method had just small changes since then [LS06] [Kan92]. Process Mapping is a graphical technique that helps in the visualization of work processes. It documents how a production process performs its work. All work or purposeful activity can be viewed as a process.

Process mapping traces a sequence of events of a product's single item within a chart. The chart presents a sequence of events from the item's perspective. By doing so, it facilitates the identification of areas for streamlining the work, reducing defects and improving operations. The chart may include additional information such as cycle time, inventory and equipment information. Although it doesn't represent information flows, they can be shown. The chart itself contains no information about the location of the events, what equipment maybe used or what workers maybe doing [LS06].

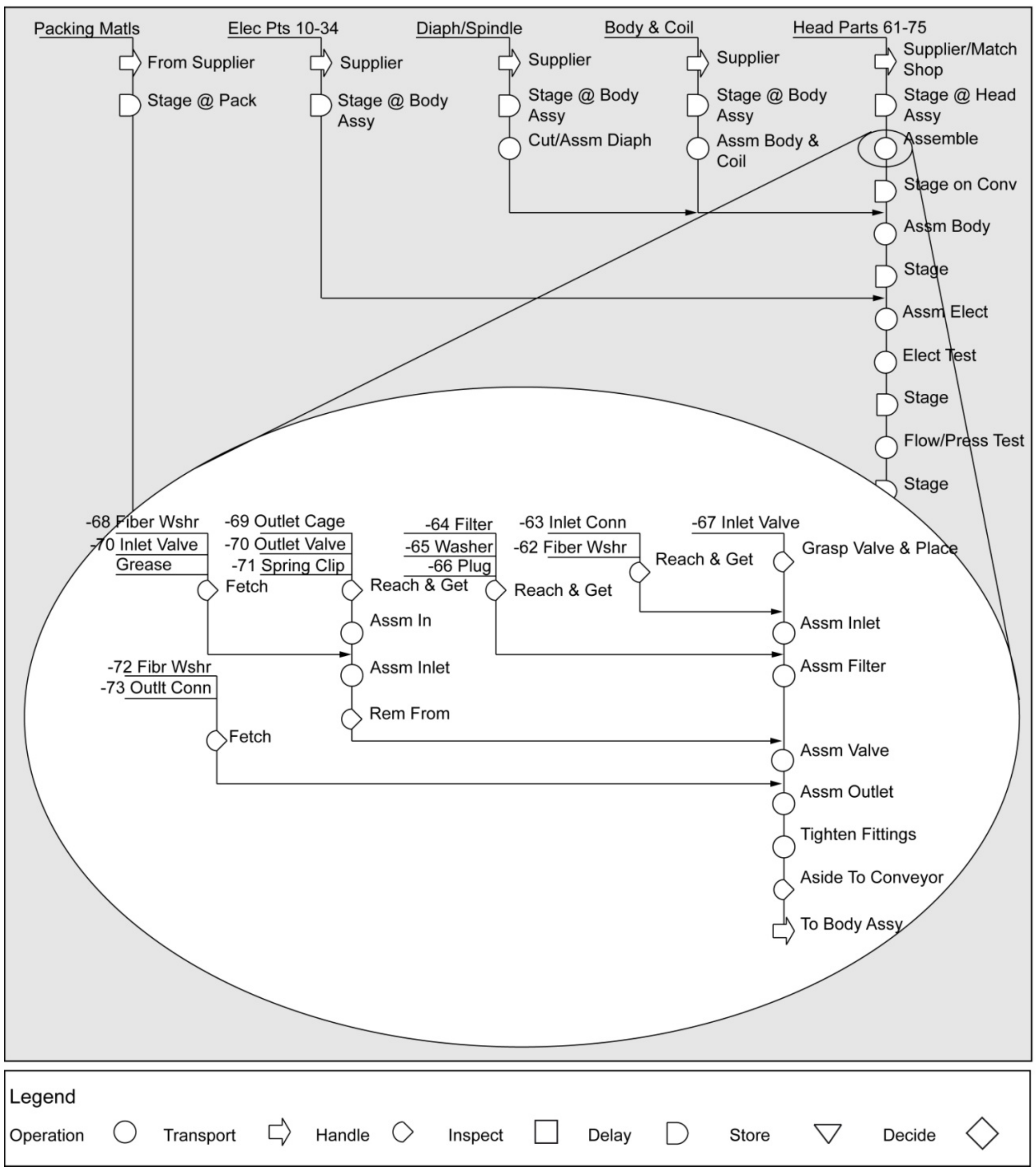

Figure 3-8: Process mapping example, the process within a process, adapted from [LS06]

Figure 3-8 presents an example of the use of process mapping. The figure presents the depiction of an assembly process example at the workcell level (gray section). The process is called “Head assembly”. The main assembly sequence is the one which starts with the *Head Parts 61-75* (located on the top right side of the gray section). Different components are assembled to this component, first the *body & coil* component, then the *diaph/spindle* component, followed by the *elec. pts. 10-34* component and the *packing matls* component. One of the operations of this process is expanded into more detail at the workstation level (white section). In the following paragraphs a brief description of the map’s lines and symbols will be given.

Line conventions: *Input lines* present components or information that enter the process, *merge lines* are used to indicate that a physical merge occurs within the main assembly line, *sequence lines* are used to represent the sequence of events while *dashed lines* are used to distinguish relevant information within the chart.

There are seven symbols (see legend in Figure 3-8):

Operation. These events transform a product.

Transport. Move a product a significant distance. In combination with the “Delay” symbol indicates a queue.

Handle. For sorting rearranging, repacking or for short moves.

Inspect. Examination of the product to determine if previous work is correct.

Delay. Situations that prevent the next process event.

Decide. Split processes into different sequences depending on some criteria.

Store. Presents physical products in an official storage location.

Process mapping can depict many levels of detail; the **macro-level** is useful for plant layout and workflows, at this level, the map is used to simplify the movements between departments of a company or well to develop part families. The **micro-level** is useful for the workcell design. In this type of map is required a detailed breakdown of the events. The **sub-micro-level** is used for the detailed design of workstations [LS06].

Evaluation of Process Mapping

Process Mapping is a visual modelling language used to depict process sequences. The symbols used within this visual technique are simple and intuitive. It presents many levels of detail and the modelling rules are very simple. Even with limited experience, a user can discern what the map represents. Nevertheless, process mapping is effective for only a single product or a group of products with high similarity, among other things, because the process maps can become quite large depending on the number of product’s parts and complexity. Speaking about the domains of a manufacturing system, this technique is focused on the process field (sequence of activities). For this reason, this technique neither considers the interrelationships between the different domains, nor supplies a domain spanning specification of the system. Furthermore, rather than a new system's conception, the focus of this technique is the visualization and documentation of existing work processes for its further analysis and improvement. It is used as a communication medium. No extensions are allowed.

3.1.7 Value Stream Mapping

The Value Stream Mapping (VSM) is a graphical technique oriented to Lean Manufacturing, that helps in the visualization of the work processes. Its original name is *material and information flow analysis*, and comes from the Toyota production system's tools [Bec05, Pg.116]. VSM consists of a set of symbols and steps that illustrate the material and information flow in manufacturing systems. It is used to describe in detail how the system works. These maps are used to document the current process as well as define the process [LS06].

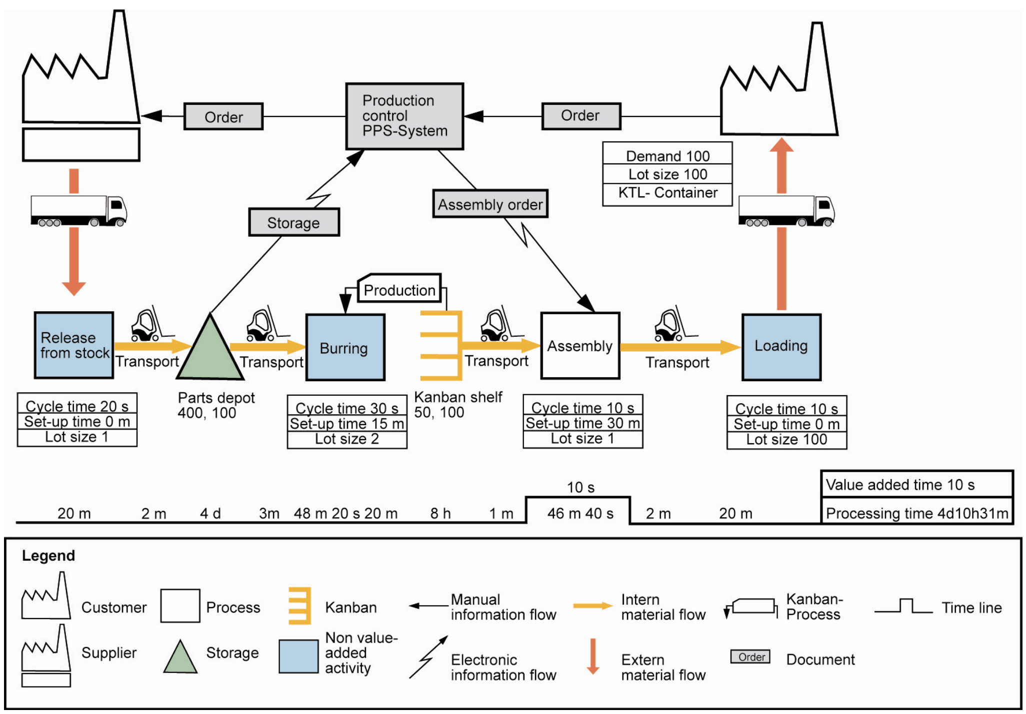

Figure 3-9: Value Stream Mapping example, adapted from [Bec05, Pg. 117]

Figure 3-9 presents the depiction of the material and information flow between a customer and its supplier. The depicted process represents the production processes and actuators of the systems with different symbols. The process starts with the customer's order (top-right corner of the map) and finishes with the merchandise's shipping to the customer.

For drawing a VSM, a product's production path from customer to supplier must be followed. Afterwards, a visual representation of every process in the material and information flow is drawn. Then, the diagram is analysed and the waste is identified. Finally, a "future state-map" which shows how value should flow can be drawn [RSW99].

VSM reflect a broad view of the process, usually from external supplier to external customer. Extended Value Stream Maps (EVSM) takes an even broader view and incorporate two or three layers of suppliers and distributors. These maps are very flexible, they could be implemented for different situations and purposes, as well as with different levels of detail. VSM goal is the elimination of *waste*. *Waste* are activities that are invisible for the customer, for example support functions like scheduling, transport, etc. Value-added and non-value added processes are terms used to identify waste [LS06] [RSW99].

The technique is oriented to high-volume, low variety production, with few components, subassemblies and dedicated equipment. The technique's symbols and conventions represent information flows, material, storage and specific Lean's techniques like Kanban. *Information symbols* represent scheduling, forecast and similar information flows that affect production. *Material* and *process symbols* represent process, equipment, departments, storage and features related directly to material [LS06].

Evaluation of Value Stream Mapping

VSM is a flexible visualization tool oriented to the Toyota version of Lean Manufacturing, it examines the physical system's processes and interconnections. Due to this, many symbols correspond to specific Toyota techniques (e.g Workcells). This fact may lead the user to employ these techniques even when they are inappropriate, due to the symbology's influence [LS06]. This specification is good for describing the processes on highly focused factories with a narrow family of products. For high variety and low-volume factories, this specification is cumbersome, due to the information amount. Additionally, the user requires training on Lean Manufacturing's concepts and symbols.

The specification was not designed for the manufacturing system conceptual design. It is focused on the analysis of *existent* processes and the *waste* identification. It allows the aggregation/adaptation of symbols. It is used as a communication medium.

3.1.8 Assembly priority chart

An assembly priority chart is a directed acyclic graph. According to the NIST[14], a directed acyclic graph is a directed graph with no path that starts and ends at the same vertex [NIS04-ol]. Applied to the manufacturing domain, this chart is a graphic representation of precedence and sequence relationships between the subprocesses of a product's manufacturing process. It is used to illustrate possible working operations (Figure

[14] The *National Institute of Standards and Technology* is a federal technology agency that develops and promotes measurement, standards, and technology.

3-10). These operations or activities are represented by a node and their interdependencies are represented with lines between the nodes [Wes06]. Each operation can not be sub-divided. It is an atomic action of a process that has no externally visible substructure. An operation is a basic process abstraction, for this reason, each operation must be completely fulfilled by a resource. Assembly priority charts are basically used to structure assembly processes and, under consideration of restrictions and the precedence relationships, it is possible to work the chart out and assign the operations suitable resources [REF87].

To create this chart, it is possible to start from the single parts of the product or from the final assembly. The principle procedure to create this chart is:

- The initial operations are registered in the chart at the left side
- Each operation is represented by a node. The nodes are linked with other nodes according to the process sequence
- Each operation has an identifier, an execution time and a brief description of the task
- Parallel operations are possible
- The amount of parallel branches increases with the assembly progress

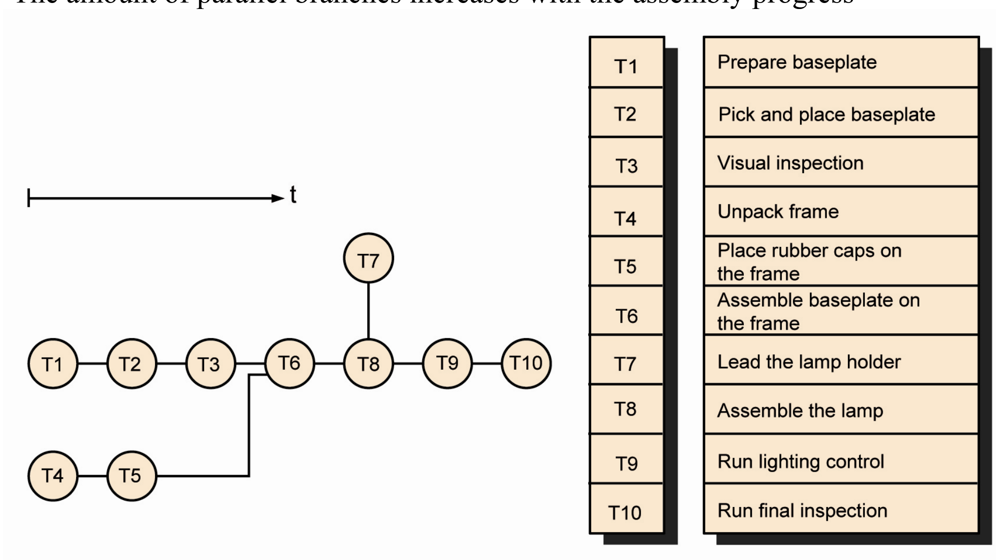

Figure 3-10: Example of an assembly priority chart. The chart explains the assembly of a car instrument [REF87, Pg. 157]

The chart's development in a logical sequence presents advantages, like [Wes06]:

- Presenting the shortest path between operations
- Aggregation of operations within a working station

- Facilitate the operation flow and the structure of the system
- Allow the single operation's timing calculation

Evaluation of assembly priority charts

Assembly priority charts are a visual modelling language used to depict process sequences. Neither is designed for the manufacturing system conceptual design nor for its development. It has simple symbols and can be used as an aid for the selection of suitable process sequences, but more detailed information (e.g. shape) of the system is not considered. It was not designed to be used as a communication medium. Extensions are not allowed.

3.1.9 VDI-Guideline 2860

The VDI-Guideline 2860 for assembly and handling contains terminology, definition and symbols for assembly and handling operations, handling functions and handling units. This guideline provides symbols to neutrally specify handling processes. Figure 3-11 presents an example.

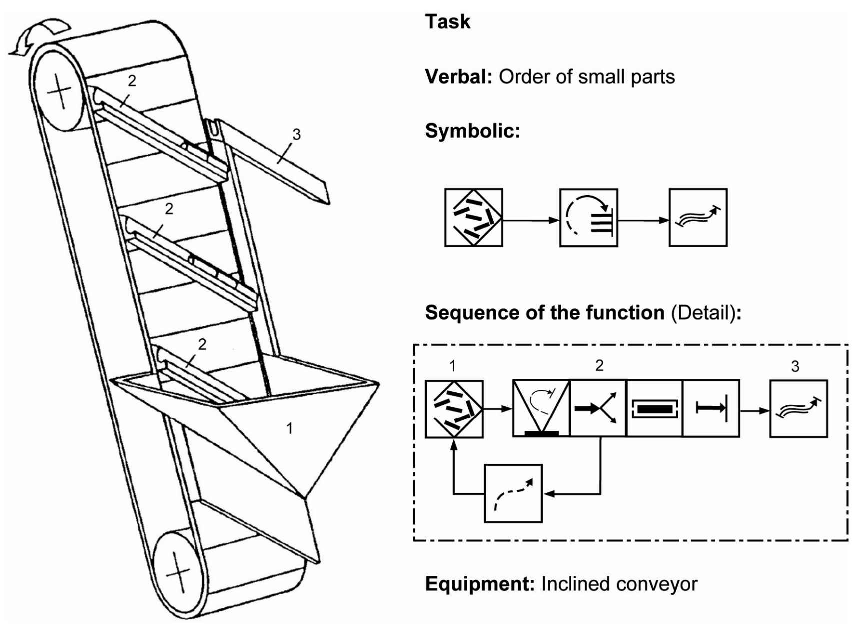

Figure 3-11: Example of the handling symbols use [VDI 2860, Pg. 10]

The *task* is "order of small parts". A neutral solution is depicted with the aid of the guide's symbols (see Figure 3-12). For a detailed description, elementary functions are used (*sequence of the function*). The depicted task can be fulfilled by, for example, an inclined conveyor. Elements of the conveyor (marked with numbers in Figure 3-11) are represented with symbols on the *sequence of the function*.

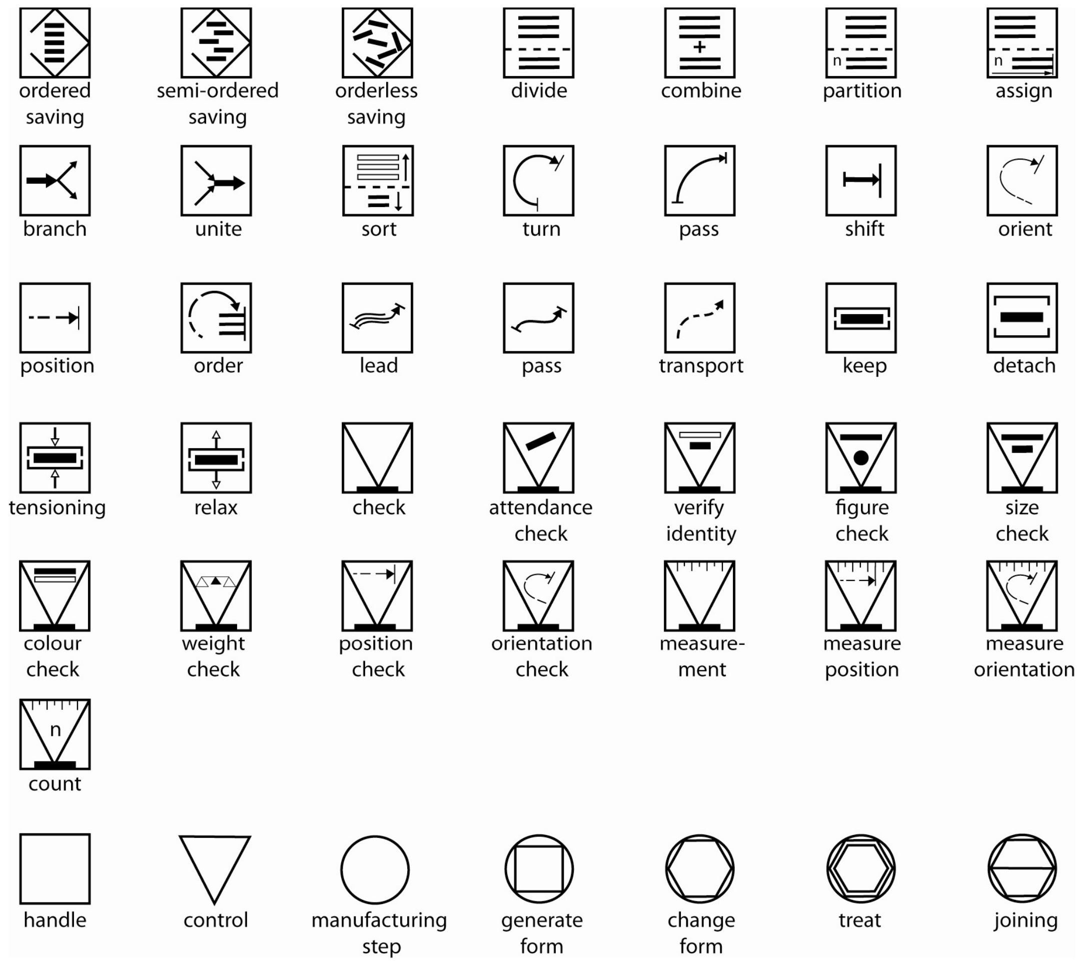

Figure 3-12: Summary of the VDI2860 symbols [VDI 2860, Pg. 16]

Handling has five partial functions (store, modification of the quantity, move, assure, control). For a refined degree of detail, seven elementary functions (divide, join, rotate, relocate, to hold, unfix, examine) are used. For the description's concretization, the individual elements can be complemented with characteristics and quantitative data. The functions and its symbols can be used to describe the sequences within technical facilities (equipment). It is also possible to model manufacturing steps and inspection steps on an abstract level. The manufacturing steps are based on the main groups of the DIN 8580 Manufacturing Processes. By the linkage of single steps, manufacturing sequences are modelled.

Evaluation of VDI 2860

The VDI 2860 provides symbols for the depiction of assembly and handling operations. The main idea is to provide a way to neutrally describe an assembly/handling task, nevertheless manufacturing process can also be depicted. This technique is focused on the description of processes and operations. It has no views or hierarchies and its symbols are quite abstract. This technique does not provide the necessary resources to depict a domain spanning specification of the system. It can be used as an aid for the neutral depiction of handling sequences, or for the selection of suitable process sequences, but more detailed information of the system is not considered. Manufacturing processes are insufficiently depicted. It was not designed to be used as a communication medium. Extensions (expandability) are not allowed.

3.1.10 VDI/VDE Guideline 3682

This guide arose from the necessity to model technical processes in engineering terms. It provides information about formalised process descriptions from an object-oriented description approach. The formalised process description consists of a graphical representation of the process (process-oriented) (see Figure 3-13), and its information model of process objects and connections (information-oriented) [VDI/VDE 3682].

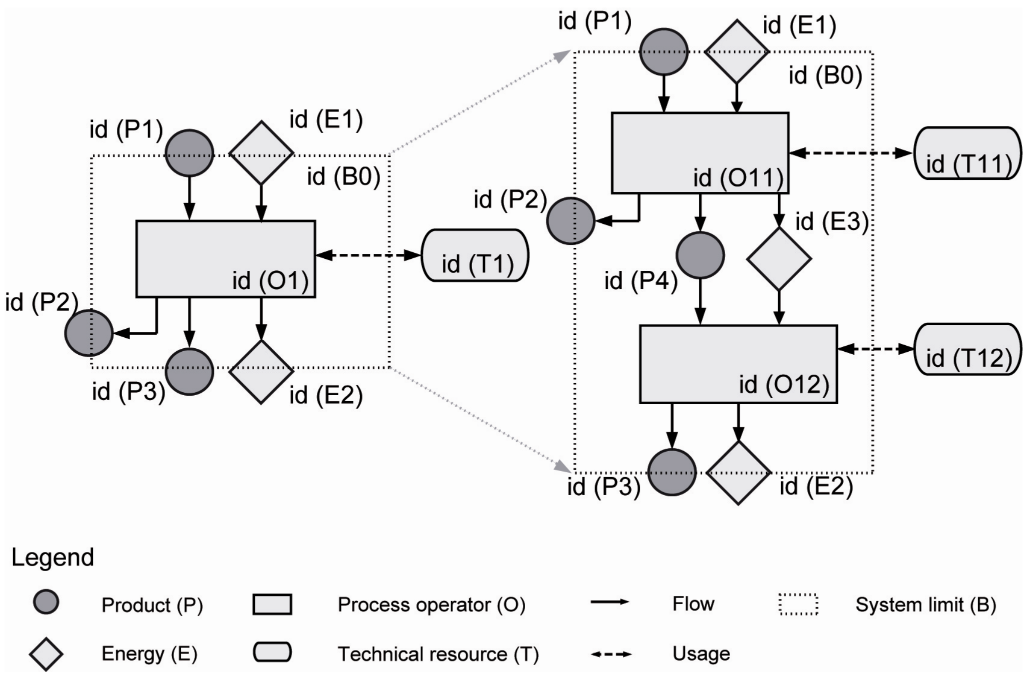

Figure 3-13: Decomposition of a process [VDI 3682, Pg.8]

The **graphical representation of the process** consists on a grid in which **states (elements)** and **process operators** are united by directed **edges**. The involved elements change from a *status ante operationem* to a *status post operationem* when the element goes through the operators. The process description is based on a **transformation principle** [VDI/VDE 3682, Pg.6]. The process is depicted graphically by a few symbols with a process-oriented decomposition. Figure 3-13 presents the seven symbols used in the graphical process description and an example of its use [VDI/VDE 3682].

The essential symbols are product, energy, process operator and flow. Secondary symbols are the technical resource and its usage as well as the system limit. **Products** and **energies** are transformed into new products and energy by a **process operator**. The actuators of the process operator are the **technical resources.** The **usage** arrow represents the use of the resource and the **system limit** is used to delimitate the analysed system [VDI/VDE 3682].

In the **information model**, definition of attributes of the process objects (process-oriented) and management of the attributes (information-oriented) are addressed, as well as export and import procedures with CA systems (information-oriented). The information model is primarily depicted by UML classification diagrams [VDI/VDE 3682].

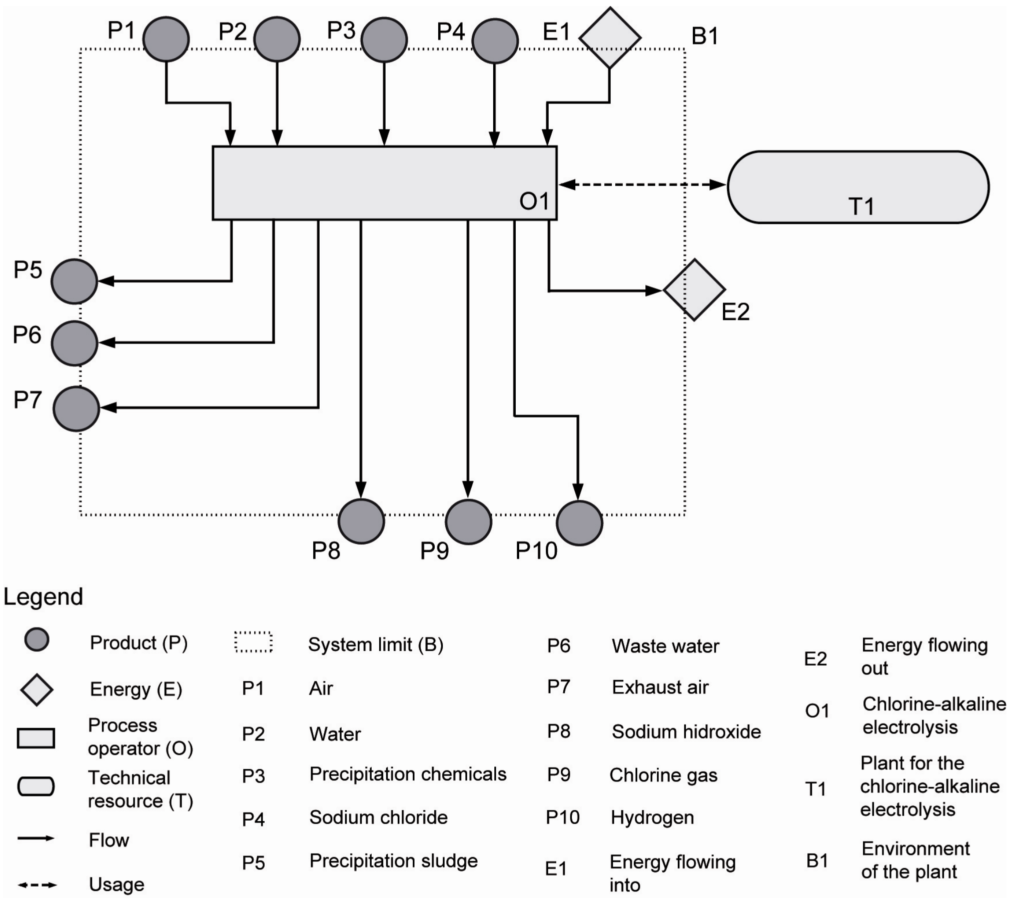

Figure 3-14: Decomposition of a process, taken from [VDI 3682, Pg.20]

Figure 3-14 presents a representation of an application example, the *chlorine-alkaline electrolysis*. The products and energies which flow into and out of the process operator “chlorine-alkaline electrolysis” (O1), the technical resource used (the plant complex) as well as the system limit are depicted within the diagram.

In accordance with the schematic, the process operator *chlorine-alkaline electrolysis* (O1) transforms the products *air* (P1), *water* (P2), *precipitation chemicals* (P3), *sodium chloride* (P4) and the energy *energy flowing into* (E1) into the new products *precipitation sludge* (P5), *waste water* (P6), *exhaust air* (P7), *sodium hydroxide* (P8), *chlorine gas* (P9), *hydrogen* (P10) and the energy *energy flowing out* (E2). The process operator implements this conversion with the help of the technical resource *plant for the chlo-*

rine-alkaline electrolysis (T1). The system limit of the process is the *environment of the plant* (B1).

Evaluation of VDI/VDE 3682

The guideline presents a simple concept to model technical processes in engineering terms. A simple symbolic is presented with a view-oriented decomposition approach. The description is simple, neutral and easily understandable. The UML based information model of the **formalised process description** is well structured. It presents the objects of the process and their connections with an object-oriented approach.

The guideline presents a practical, graphical, and descriptive language for process-oriented specialists. The symbols are simple and intuitive. It presents many levels of detail and the modelling rules are relative simple.

Nevertheless, the **graphical representation of the process** is based on the **transformations** of "**states**". The **product** and **energy elements** are used for describing these "states". This fact implies that it is not possible to depict "storage" processes, because no **transformation** occurs. Furthermore, the representation depicts series of **transformations** no **sequences**. Due to its approach, neither is possible to depict a resources sequence, nor aggregation ("usage") of resources, nor processes sequences.

This technique does not provide the necessary resources to depict a domain spanning specification of the system, because this technique is focused on the processes' description. Furthermore, **rather than a new system's conception**, the focus of this technique is the visualization and documentation of **existing processes**. It was designed for the process description, but considering that the users will be people from different fields. For this reason, the used nomenclature was carefully chosen, to facilitate its comprehension. No extension is allowed.

3.1.11 IDEF

IDEF[15] is the name of a group of methods used for enterprise modelling and analysis. Originally they were developed for the United States Air Force. The idea was to develop a function modelling method for analyzing and communicating the functional perspective of a system. There are six IDEF methods. The most commonly used are IDEF0 and IDEF3 [IDE10a-ol].

IDEF0 was designed for the functional modelling (**function-oriented approach**). It was derived from a well-established graphical language, the Structured Analysis and Design Technique (SADT). IDEF0 is based on a cell modelling graphic representation. This method models elements controlling the execution of a function, the actors performing the function, the objects or data consumed and produced by the function, and

[15] Integrated DEFinition methods

the relationships between business functions. IDEF0 allows an organised representation of activities and the important relations between these activities in a non-temporal approach [IDE10a-ol] [IDE10b-ol].

IDEF0 is a communication and analysis tool based on a functional model (see Figure 3-15). This functional model has five elements, so called ICOMs (*I*nputs, *C*ontrols, *O*utputs, *M*echanismus):

- **Activity**, represented by boxes
- **Inputs,** represented by arrows
- **Outputs,** represented by arrows
- **Controls,** represented by arrows
- **Mechanismus,** represented by arrows

The *controls* or constraints, regulate the activity, and the *mechanismus* carry out the activity. Additionally, the model can be decomposed into more detailed levels of analysis [IDE10a-ol] [IDE10b-ol].

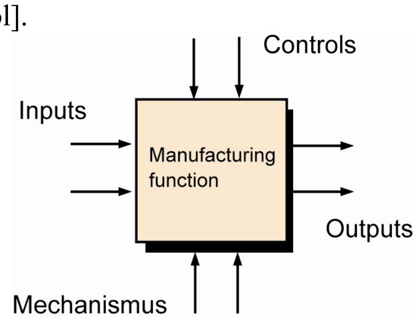

Figure 3-15: IDEF0 Box and Arrow graphics [IDE10a-ol]

IDEF0 does not support the specification of processes. For this reason, IDEF3 arose. **IDEF3** was designed for the process modelling: It allows a detailed description of the specific logic or timing associated with processes. It captures the workflow of a business process by means of process flow diagrams. IDEF3 presents a task sequence for processes, allows the description of different scenarios for performing the same business functions, and enable the analysis and improvement of the workflow. A main difference between IDEF3 and the other IDEF methods is that IDEF3 creates *descriptions*, by means of diagrams and text while other IDEF methods produce *models* [IDE10a-ol] [IDE10c-ol] [IDE10d-ol].

In the context of IDEF:

> *„The term* ***description*** *is used as a reserved technical term to mean records of empirical observations; that is, descriptions record knowledge that originates in or is based on observations or experience. The term* **model** *is used to mean an idealization of an entity or state of affairs. That is, a model constitutes an idealized system of objects, prop-*

erties, and relations that is designed to imitate, in certain relevant respects, the character of a given real-world system. Frictionless planes, perfectly rigid bodies, the assumption of point mass, and so forth are representative examples of models" [IDE10d-ol, Pg. 4]

IDEF3 captures the behavioural aspects of an existing or proposed system. It has two description modes, process flow and object state transition network. The **process flow** description (Figure 3-16) captures knowledge of the working in an organisation (e.g. sequence of manufacturing processes). The **object state transition network** description (Figure 3-17) summarizes the permissible transitions that an object may undertake throughout a particular process.

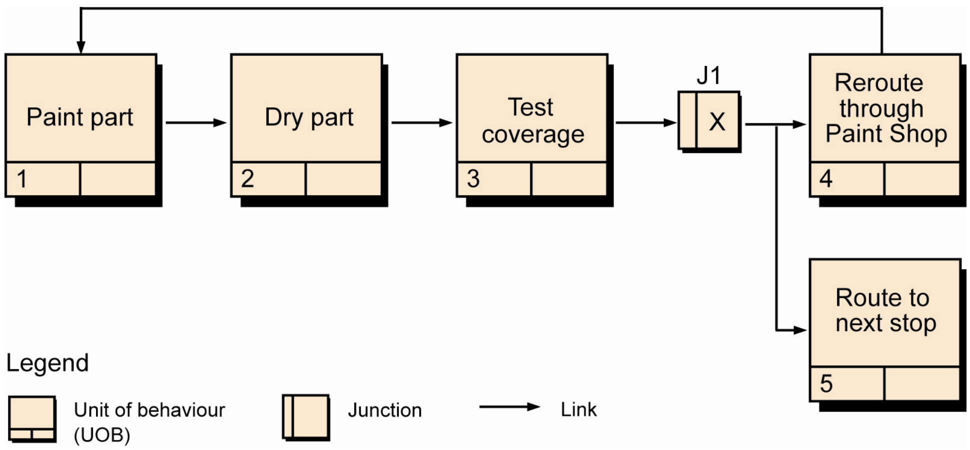

Figure 3-16: IDEF3 Process Description Diagram [IDE10c-ol]

Figure 3-16 presents a process description where a painting and inspection processes are depicted. This example is the graphical representation of the scenario (story) told by a paint shop supervisor when asked to describe: "What goes on in the primer shop?"

„Parts (of a product) enter the shop ready for the primer coat to be applied. We apply one very heavy coat of primer paint at a very high temperature. The paint is allowed to dry in a bake oven after which a paint coverage test is performed on the part. If the test reveals that not enough primer paint has been sprayed on the surface of the part, the part is re-routed through the paint shop again. If the part passes the inspection, it is routed to the next stop in the process" [IDE10c-ol].

The activities described in the story, appear as labelled boxes in Figure 3-16. The labelled boxes can describe activities, processes, events, etc. The IDEF3 term for elements represented by boxes is a **Unit Of Behaviour** (UOB). The arrows (**links**) tie the UOBs together and define the logical flows. The smaller boxes define junctions. **Junctions** are used when processes split or multiple paths merge [IDE10d-ol] [IDE10c-ol].

The **Object State Transition Network** (OSTN) description (Figure 3-17) summarizes the permissible transitions that an object may undertake throughout a particular process.

Object states and state transition arcs are the key elements of an OSTN diagram. **Object states** are represented by circles and **state transition arcs** are represented by the lines connecting the circles (links).

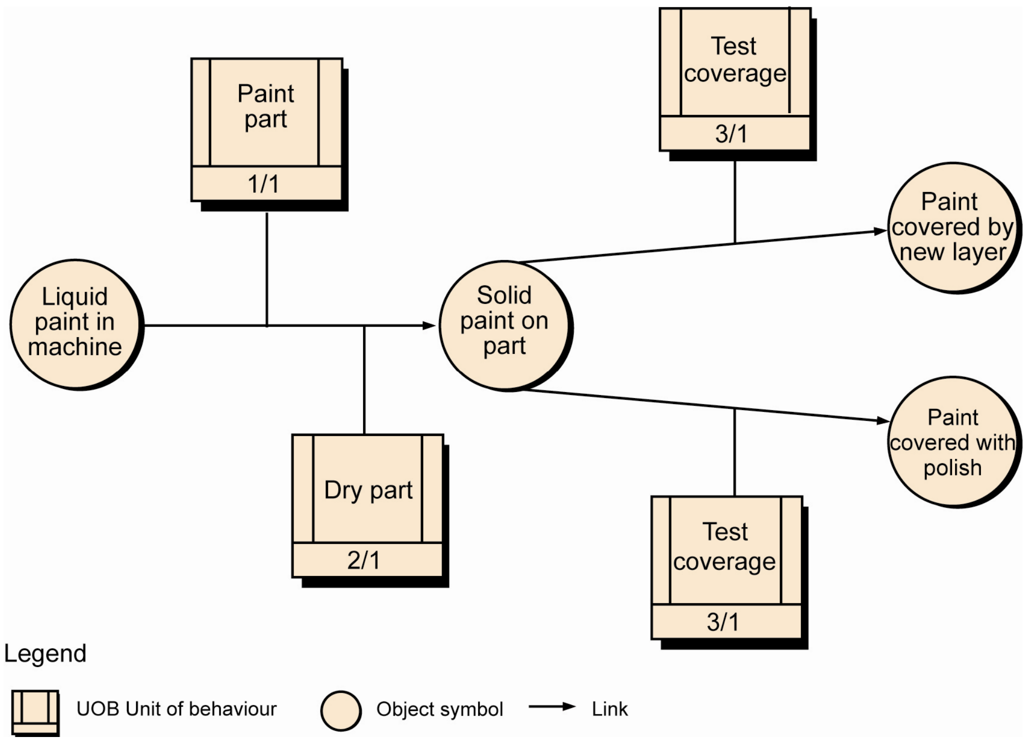

Figure 3-17: IDEF3 Object State Transition Network Diagram [IDE10c-ol]

An *object state* is defined in terms of the facts and constraints that need to be true for the continued existence of the object in that state. Entry conditions specify the requirements that need to be met before an object can transition into a state. Exit conditions characterize the conditions under which an object can transition out of a state. The constraints are specified by a simple list of property and value pairs or by a constraint statement [IDE10c-ol] [IDE10d-ol]. *State transition arcs* represent the allowable transitions between object states. The participation of a process in a state transition between two object states can be represented by attaching a UOB referent to the transition arc between the two object states [IDE10c-ol] [IDE10d-ol].

Figure 3-17 presents an example of an object state transition network. The diagram captures the object-centered views of the processes described in Figure 3-16. The diagram summarizes the allowable transitions (object state), their sequence and the processes involved with these transitions (UOBs).

Evaluation of IDEF

IDEF0 is a communication and analysis tool based on a functional model. It is designed for the system's analysis, rather than for its conceptual design. IDEF0 allows an organised representation of **activities** and the important relations between these activities in a non-temporal approach. IDEF0 was not designed to depict a domain spanning specification of a system.

The description of the system's activities can easily be refined, by decompositions, until the model is as descriptive as required. IDEF0 diagrams are based on simple boxes and arrows. Nevertheless, the obtained models are often so concise that only domain experts or people that have been involved on the model development can understand them [IDE10a-ol]. The IDEF0's abstraction is away from timing, sequencing, and decision logic. However, the users tend to interpret the models as structures that represent a sequence of activities. No extension is allowed. For detailed descriptions of the specific logic or timing associated with the activities, it is required the IDEF3 Process Description Capture Method.

IDEF3 development arose from the necessity to speed up the process of business systems modelling [IDE10d-ol]. However IDEF3 offers a good platform for capturing and analyzing processes of an existing or proposed system.

The IDEF0 and IDEF3 methods provide support for the modelling (IDEF0) and description (IDEF3) of several architectural views of systems in general. This implies that an adaptation must be made before using them for the manufacturing system description and development. Furthermore, there are no communication mechanisms between the models and the descriptions. This hinders the visualisation of possible interrelated elements of an architectural system. A switch between views is not possible.

3.2 Need for action

Based on the comparison between the requirements listed on Chapter 2.4 and the characteristics presented by the specification techniques expounded on Chapter 3.1, the following evaluation (Figure 3-18) was derived:

The requirements are:
- Totally fulfilled ●
- Partially fulfilled ◑
- Not fulfilled ○

Approaches / Requirements	Holistic description of a manufacturing system	Description of a manufacturing system at an early stage	Consistent semi-formal specification technique	Domain spanning specification technique	Support the iterative process of the conceptual design	Reduction of complexity	Expandability	Communication medium
GRAPES	◑	◑	●	◑	◑	◑	○	◑
Integrative specification of the product's and production system's conceptions	◑	◑	●	●	●	●	●	◑
SysML	◑	◑	●	◑	◑	◑	○	◑
Manufacturing process model	◑	◑	●	◑	◑	◑	○	◑
REFA's symbology to depict flexible manufacturing systems	◑	◑	●	○	○	◑	○	○
Process mapping	◑	◑	●	○	○	●	○	●
Value stream mapping	◑	◑	●	○	○	◑	●	●
Assembly priority chart	◑	◑	●	○	○	○	○	○
VDI-Guideline 2860	◑	◑	●	○	○	○	○	○
VDI/VDE-Guideline 3682	◑	◑	●	◑	●	●	○	◑
IDEF	◑	◑	●	○	○	●	○	◑

Figure 3-18: Evaluation of the analysed specification techniques

In the following paragraphs, a brief justification for the above displayed evaluation is presented.

Requirement R1 - Holistic description of the manufacturing system's principle solution

None of the examined specifications totally fulfilled this requirement. Some of the specifications were designed for systems in general. For this reason, an adaptation needs to be made in order to use them (e.g. SysML, GRAPES). Other specifications describe a manufacturing system, but not completely. In some cases, just value-added processes are considered (e.g. MICHELS), in others the description is really abstract (e.g. REFA) or it doesn't consider the system with an holistic approach. It just focuses on one domain (e.g. the *process* domain, like in Process Mapping and Value Stream Mapping).

Requirement R2 - Description of a manufacturing system at an early stage of the product development process

None of the examined specification techniques totally fulfilled this requirement. In some cases the fulfilment of this requirement depends on the fulfilment of the requirement 1. If a specification which was designed for systems in general is not suitable adapted, then it is not going to be able to describe a manufacturing system at an early stage. In other cases, the specifications were focused on the process description. They were designed for the analysis, not for the conception and development. For this reason, these specifications "model" what already exists, and are not able to model non-concrete elements.

Requirement R3 - Consistent semi-formal specification technique

All the analysed specification techniques fulfilled this requirement.

Requirement R4 - Domain-spanning specification technique

Most of the analysed specifications were designed for the process description and analysis (e.g. REFA, process mapping, value stream mapping, assembly priority chart, VDI2860, IDEF). In the case of GRAPES and SysML, the fulfilment of this requirement depends on the adaptation that should be made to model manufacturing systems. Other specifications (e.g. Manufacturing process model and VDI/VDE 3682) consider two or three views from the four required ones (requirements, process sequence, resources, shape). Furthermore, the views are not considered in a balanced way, commonly, the focus was on the process field. The integrative specification (MICHELS) was the only specification that fulfilled this requirement.

Requirement R5 - Support the iterative process of the conceptual design's development and its concretization

Most of the specifications (e.g. REFA, process mapping, value stream mapping, assembly priority chart, VDI2860, IDEF) do not support the conceptual design. These were designed for the description and analysis. In some cases, they do not offer different views of the system or the possibility to concretize the depicted objects. Some specifications support the iterative process of the conceptual design, (e.g. SysML and GRAPES), nevertheless, because of the amount of diagrams and elements used on these specifications is quite easy to lose the overall view. The VDI/VDE3682 and the integrative specification (MICHELS) fulfil this requirement.

Requirement R6 - Reduction of complexity

Some of the specifications (e.g. GRAPES, SysML, manufacturing process model, REFA, value stream mapping) partially fulfil this requirement using different views, aggregation/ decomposition, etc. Other specification techniques reduce the complexity by using an easy language, few rules and few layers. The VDI2860 does not fulfil this requirement because it has no views or hierarchies and its symbols are quite abstract. The assembly priority chart offers no resources for taming the complexity, it was made for the depiction of sequences.

Requirement R7 - Expandability

Just two specification techniques (MICHELS and Value Stream Mapping) allow the addition of symbols. This requirement is necessary for a holistic description of a manufacturing system. Depending on the required model's degree of concretization, it may be necessary to complement the specification with different modules that allow the specification of concrete elements (e.g. sensors).

Requirement R8 - Communication medium

Just two specifications (Process mapping and value stream mapping) can be used as a good communication medium, because of its easy structure and the amount of information that they contain. These specifications were designed for the team work, they are easy to understand, and were developed for the practical use. Some of the here analysed specifications partially fulfil this requirement (e.g. GRAPES, MICHELS, SysML, manufacturing process model, VDI/VDE3682, IDEF) because their structures are complex, the amount of information that they contain and the relationships between their views/layer/diagrams are not so easy to represent, and what is more important, to keep in mind. Other specifications are easy to understand, but they do not contain enough information to be used as a representative communication medium (e.g. assembly priority chart, REFA, VDI2860).

Summary

The here presented specification techniques are suited for different situations and different purposes; they also vary in their level of detail and abstraction. Most of the specifications here analysed were designed to model systems in general. Models are simplified depictions of the reality, therefore any specification technique presents just certain aspects of the reality ignoring other aspects. Specifications do not have to be complete accurate to be useful. The important point is to be aware of the possible inaccuracies and the specification's focus, to work accordingly. Modelling is only a tool to assist with certain aspects of developing a comprehensive and effective manufacturing system. The here analysed specifications were not designed specifically for manufacturing systems at early stages of the product development, nevertheless they present different approaches that combined or adapted may be a suitable base for a specification technique focused on manufacturing systems.

4 A specification technique for the conceptual design of manufacturing systems

The state of the art's results endorse the necessity of a language for the conceptual design, development, description and documentation of the principle solution of manufacturing systems. This language must allow a basic description of the manufacturing system. A description which specifies the essential structure and functionality of the system, enough concrete to understand the overall system, but still abstract enough to allow changes and modifications.

In the following chapter, a specification technique for the description of the principle solution of manufacturing systems is presented. The starting point for this specification is the work of FRANK, *Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme* [Fra06]. Continuing along with this conception, the following work provides a language for the conceptual design, development, description and documentation of the principle solution of manufacturing systems at an early stage of the product development process. To achieve this goal, an holistic and integrative semi-formal specification technique was developed. This specification uses some elements and concepts from [Fra06], [Mic06], [Ste97], SysML, GRAPES and IDEF. Nevertheless, the here presented specification is not to understand as an extension of the mentioned literature, but rather constitutes a new language.

4.1 Basic idea – A specification based on a coherent system of cross-linked partial models

Manufacturing systems are complex systems. For taming the system's complexity, the system needs to be divided into aspects (also known as views). Through this model construction, isolated abstractions of the systems are created, enabling a comprehensive description of the system.

The term **manufacturing system's principle solution** (see Chapter 2.2.2.7) was defined as a first rough description of a manufacturing system. The system presents a **collection of interrelated resources** that perform a set of activities (**processes**). These activities convert goods (inputs), according to a set of **requirements** and needs, into a variety of products, services and information (outputs) in the most possible cost effective way.

Based on the *problem definition* (see Chapter 2.3) and the *manufacturing system's principle solution* definition, manufacturing systems can be divided in 4 interrelated views. These views are: **requirements, process sequence, resources** and **shape of the manufacturing system**. As well as with the product's views, the manufacturing system's views are mapped on computer by partial models. These partial models have to be integrated to enable an holistic and integrative design. Therefore the principle solution consists, analogue to the product's one, of a coherent system of partial models (Figure 4-1).

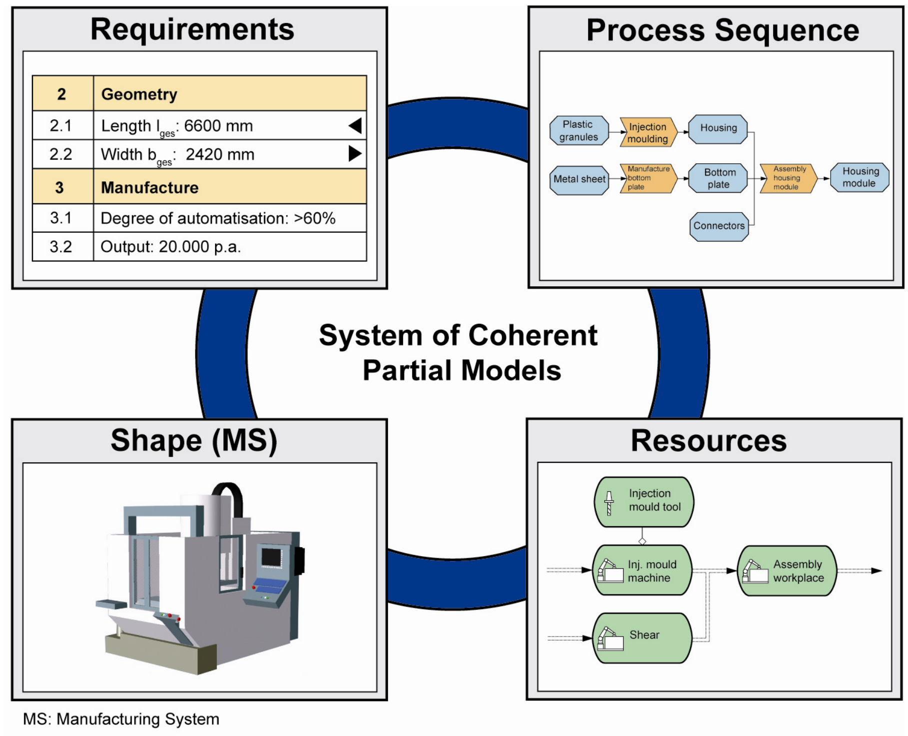

Figure 4-1: Views for the cross-functional description of the principle solution of manufacturing systems

The partial models and their interaction represent and document the product's manufacture in the course of the system's concretization. Each partial model represents *a closed structured group of individual objects* which interact with other objects of other partial models. These objects are independent objects which are related with each other. A model of a system is thus a network of objects, which are defined in terms of the relationships between each other, their internal characteristics and its behaviour.

In the following paragraphs, a brief description of the partial models is provided. The partial models (PM) are discussed in more detail in Chapter 4.4.

The **PM requirements** considers the computer-internal representation of manufacturing systems' requirements. Thus, the objects here (text) are an organised collection of requirements on the product and the manufacturing system.

The **PM process sequence** considers the computer-internal representation of the manufacturing sequence. The objects here are *elements* and *processes*. These objects are represented by means of constructs, which will be treated with more detail on Chapter 4.3.1. The *elements* represent the initial, intermediate and final state of the product, the *processes* represent the operations that have to be made in order to achieve a desired attribute within the product.

The **PM resources** considers the computer-internal representation of the processes' resources, e.g. machines and workers and the material flow that passes through them. The objects here are resources represented by means of constructs.

The **PM shape of the manufacturing system** considers the computer-internal representation of the actuators' shape. The objects here are written specifications, sketches or CAD data.

Basic concepts for modelling with the specification technique

This specification was designed for modelling the conceptual design of manufacturing systems at an early stage of the product development process. At this point, scarcely concrete information is available. Nevertheless, the information's contents increase in the course of the system's concretization. For this reason, a static model of the system, with a top-down approach (from abstract-to-concrete) was developed.

The specification technique provides a language for the **description** and **documentation** of the manufacturing system's principle solution. The *description*, or graphic representation of the system, is done by constructs, pictograms, arrows and geometric elements. The *documentation* of the system is achieved by tables, graphic sequences and shape models. The information of the product and the manufacturing system are depicted as a text, as a graphic or as a combination of both. As a result, the here presented domain-spanning specification technique for manufacturing systems allows a comprehensible, holistic and integrative representation of the system.

This specification was built up as a core specification technique with indirect expandability. Indirect expandability means that, if required, the user can extend the specification (e.g. add a view, add constructs and relationships, etc.) with modules. These separated modules may contain new objects which are needed to depict specific requirements that further concretizations of the principle solution needs.

4.2 Principle approach of the conceptual design of manufacturing systems

The conceptual design of manufacturing systems is based on the principle solution of the product. Specifically, three aspects of the product's principle solution contain relevant information for the design: *Requirements, Active structure* and *Shape* (see Figure 2-3). These three aspects mark the starting point. The partial model *requirements* not only contains relevant product requirements, but also manufacturing system related requirements, e.g. the production rate (pieces/hour) or the automation degree. The *active structure* and the *shape* describe the components which need to be manufactured and their spatial relations.

The principle process of the conceptual design of manufacturing systems by means of the partial models and their interrelationships is illustrated in Figure 4-2.

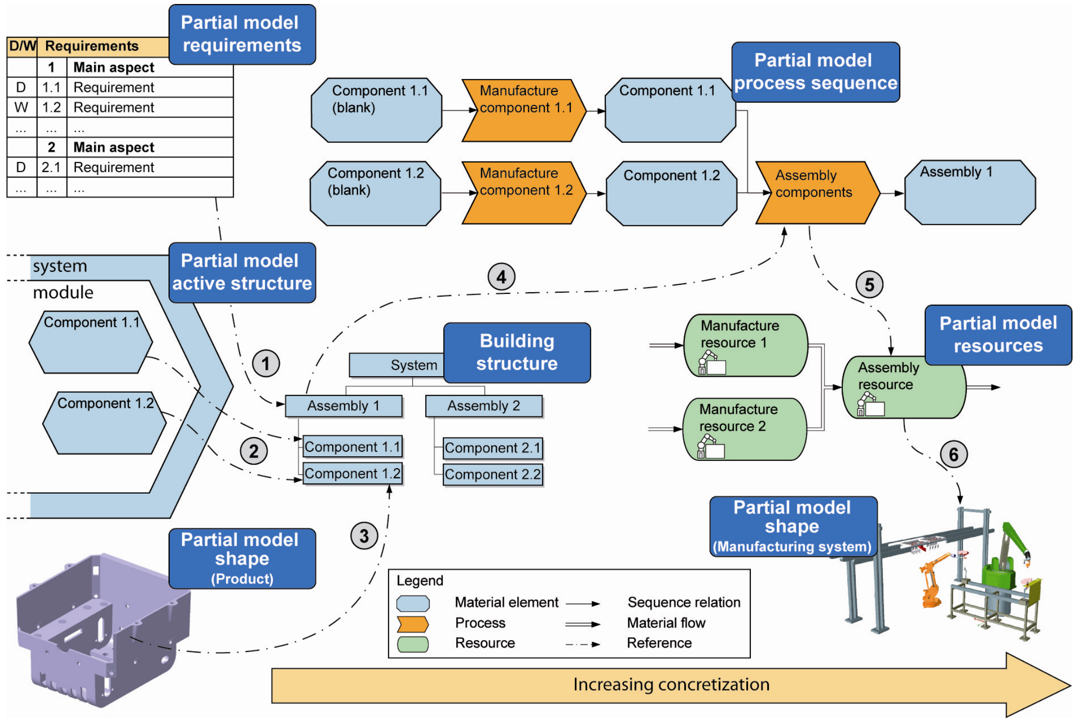

Figure 4-2: Principle approach of the conceptual design of manufacturing systems

Based on information of the product's principle solution, specifically, the information from the *requirements* (1), the *active structure* (2) and the product *shape* (3) a first *building structure* is developed. The *building structure* describes the components of the product with its spatial and logical aggregations [PBF+07]. Based on the functional and spatial correlations of *active structure* and *shape* and, if necessary, further partial models, the components and assemblies are structured under manufacture and assembly aspects [Ste07]. In this case (see Figure 4-2), the *building structure* (3) shows that the component 1.1 and 1.2 are assembled to an assembly. The two assemblies "Assembly 1" and "Assembly 2" form the "System".

The information contained in the *building structure* is used to establish a first assembly and manufacturing sequence (4). The processes as well as the product components are depicted on the partial model *process sequence*. The *process sequence* is the core of the principle solution of the manufacturing system.

Each process is described by a **manufacturing function**[16]. Manufacturing functions are solution neutral operations used to describe manufacturing process. Manufacturing functions are concretized to **manufacturing processes**[17] and these in turn, to **technologies**[18]. Thereby the interactions between the applied technologies need to be considered and alternative technology chains need to be evaluated. For that approved methods are used, e. g. the methodology for evaluating manufacturing sequences by [BKW+05].

Next, *resources* are allocated to the several processes (5). *Resources* are needed to realize the chosen manufacturing technologies [CHN02], [ZMR05]. For this allocation diverse methodologies and systematic processes can be used (see Chapter 2.2.2.5). In this case, an assembly resource is assigned to the assembly process. Resources have a *shape* (6). *Shape* can be the space needed by a resource or a concrete floor space. It might also be a first CAD model of the manufacturing system (see Figure 4-2).

The four mentioned partial models and its interrelations describe the principle solution of the manufacturing system. In the following chapters, a detailed description of the constructs and the partial models is given.

4.3 Constructs, pictograms and relationships

The specification technique consists of four partial models. Two of these four partial models are depicted graphically by means of a diagram. Diagrams provide a graphical medium that helps experts from different domains to communicate knowledge. Diagrams are depicted with constructs.

BUNGE defines a construct as a conceptual or ideal object [Bun06, Pg. 27]. The Merriam-Webster dictionary defines it as the draw of a geometrical figure with suitable instruments and under specified conditions [Mer10-ol].

In this specification three types of constructs are used: Basic constructs, pictograms, and relationships. In the following paragraphs a description of these constructs will be presented.

---

[16] A **function** is characterized by the combination of a substantive and a verb. It is a relationship between input and output variables of a system. This relationship is described as an operation. This operation is purpose oriented and solution neutral [PL08, Pg.56]. In this work, it is considered as a **manufacturing function** the terms classified in the categories “main groups” and “groups” within the VDI8580 [VDI8580].

[17] In this work, it is considered as a manufacturing process the terms classified in the category “minor group” within the VDI8580 [VDI8580].

[18] In this work, technologies are considered the last concretization of a process. A resource is the actuator of a process.

4.3.1 Basic constructs

The process sequence partial model and the resources partial model are depicted graphically by means of a diagram. A *process sequence diagram* is constructed systematically, using two basic constructs, the *element* construct and the *process* construct. A resources diagram is depicted with one basic construct, the *resource* construct. In the following paragraphs a description of these three constructs will be presented.

Element

There are two types of components within the building structure: **material elements** and **software components**. *Material elements* are defined as all raw materials, auxiliary materials, components from suppliers and trade goods, as well as raw, unfinished and finished goods [GK06]. Material elements are used to represent the different states, transitions and transformations of the product through the different stages of the manufacturing process. Material elements have *shape* and are depicted with an octagon, the graphical representation of an *element construct* (Figure 4-3).

Software components [FGK+04] include software programs and applications (e.g. Java program). Analogue to *material elements,* software components are represented by an octagon, the *element construct* (Figure 4-3). Nevertheless, due to its nature, a software component does not have *shape*, as the material elements do.

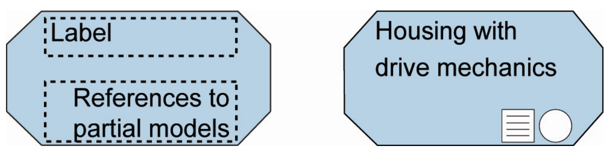

Figure 4-3: Graphic representation of an element

The *element construct* contains two information fields:

Label: In this space is written the name of the element. Commonly, a substantive and in some cases a substantive with an adjective or a description that explains the state, transformation and/or transition of the element.

References to partial models: Here are depicted pictograms which make reference to partial models. These partial models are used to define and concretize the material elements and software components. The material elements can have a reference to the *partial model requirements* and a reference to the *partial model shape*. A software component can only have a reference to the *partial model requirements*.

Process

Processes comprise all the necessary activities required to manufacture products and services, in units of personal (workforce), technique (means of labour) and organisation. It comprises manufacturing, assembly and logistic processes.

Manufacturing processes: These are the processes required to obtain a first shape or state from the formless original state to change the material properties. As manufacturing processes are considered predominantly manufacturing processes according to the standard DIN 8580.

Assembly processes: As assembly processes are considered joining processes according DIN8593-0, but also handling and control operations, like the ones of the VDI2860.

Logistic processes: In general, logistic processes include planning, design, and support of operations of procurement, purchasing, inventory, warehousing, distribution, transportation, customer support, financial and human resources. Nevertheless, during the conceptual design of the manufacturing system, there is scarce information. At the conceptual stage just two main processes are considered: transport and storage.

Each process performs or executes a transformation of the shape and /or, properties of the material element. A process can also modify the state (e.g. temperature), position (e.g. vertical) and location of material elements. In the case of software components, processes include to store, save or burn the component within "hardware" components.

Depending on its degree of complexity, some processes can be decomposed, to single processes or activities. The same principle can be used in the opposite way. Single processes can be aggregated into complex process. In Chapter 4.3.2 this theme is discussed.

Any process has at least one element as input and at least one element as output (see Figure 4-4). Each process is characterised by an operation (function, process) and additional attributes (e.g. process parameters).

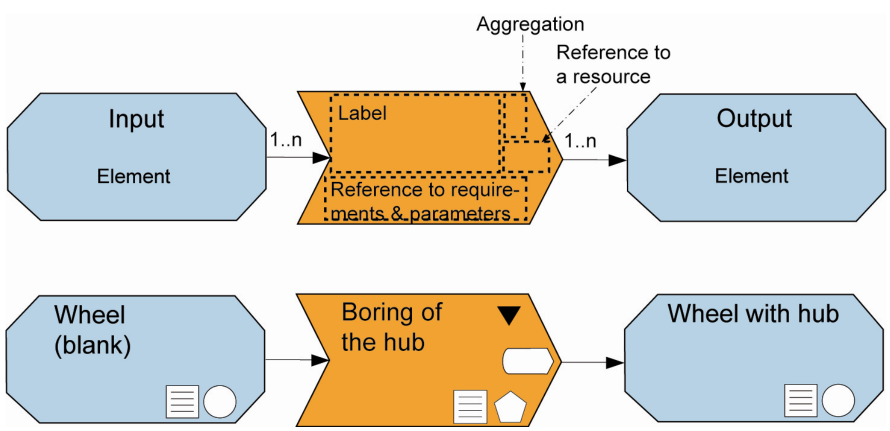

Figure 4-4: Graphic representation of a process

The process construct contains four information fields:

Label: In this space is written the name of the process. Commonly, this name consists on a verb and a substantive which explain the transformation, transition or activity to execute on the element.

Reference to requirements and parameters: Here are depicted pictograms which make reference to the partial model requirements and parameters. The parameters contain information about the process, like tolerances, technical information etc. Parameters are going to be treated with more detail on Chapter 4.3.2.

Aggregation: Here is depicted the aggregation pictogram in case that the analysed process is part of a process or in case that the process itself has subprocesses (see Chapter 4.3.2).

Reference to a resource: In case that the analysed process is related to a specific resource, a process-resource pictogram is placed in this area. This pictogram links the process construct with its related resource construct (see Chapter 4.3.2).

Resource

With resources are meant all the equipment, tools and personnel that are required for the execution of processes [DIN69902]. Resources are processes' actuators. To every process at least one resource is allocated, whereas it is possible that one resource realises more than one process.

Resources are used to depict the production structure. Each resource is joined to another resource by means of a material flow relationship (Chapter 4.3.3). Each resource acts as a resource "source" for the next resource. The starting point for any production structure is a resource from the type *storage*. Each resource is characterised by a name, additional attributes (resources parameters), and, if necessary, it can have references to partial models (to the *partial model shape*, the *partial model requirements* and/or the *partial model process sequence*). Resources can be detailed by means of aggregation relationships. Detailed information about this theme is discussed on Chapter 4.3.3. Figure 4-5 presents a resource's example.

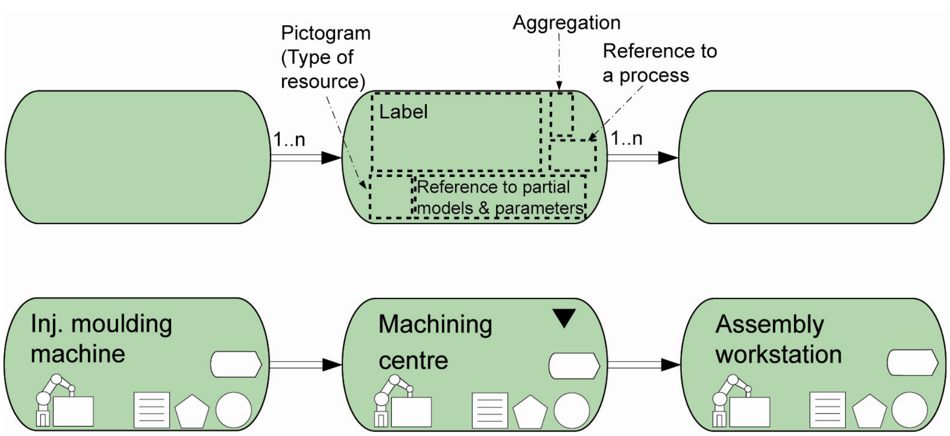

Figure 4-5: Graphic representation of a resource

The construct contains four information fields:

Label: In this space is written the name of the resource. Commonly, this name consists on a substantive which explain the type of resource, for example, worker, gripper, etc.

Reference to partial models and parameters: Here are depicted pictograms which make reference to the *partial model shape*, *partial model requirements* and parameters. The references are used to define and concretize the resource and its relations within the manufacturing system. Parameters are going to be treated with more detail on Chapter 4.3.2.

Pictogram (type of resource): There are five different types of pictograms which help to point out the type of resource: Worker, machine, tool, transport, storage. These offer an optional visual aid for the swift resource identification. Detailed information of the pictograms is presented on Chapter 4.3.2.

Aggregation: Here is depicted the aggregation pictogram in case that the analysed resource is part of another resource or in case that the resource itself has resources (see Chapter 4.3.2).

Reference to a process: In case that the analysed resource is related to a specific process, in this area is placed a process-resource pictogram, which links the resource construct with its related process construct (see Chapter 4.3.2).

4.3.2 Pictograms

According to the Oxford dictionary, a pictogram is a pictorial symbol for a word or phrase [Oxf10-ol]. According the Merriam-Webster dictionary, an ideogram is a picture or symbol used in a system to represent a thing or an idea but not a particular word or phrase for it; especially one that represents not the object pictured but some thing or idea that the object pictured is supposed to suggest [Mer10-ol]. In general, a pictogram is an ideogram that conveys its meaning through its pictorial resemblance to a physical object. In this work, the pictograms are clustered in three groups according its use (see Figure 4-6): As a reference to partial models, as visual information and indicating detailed information.

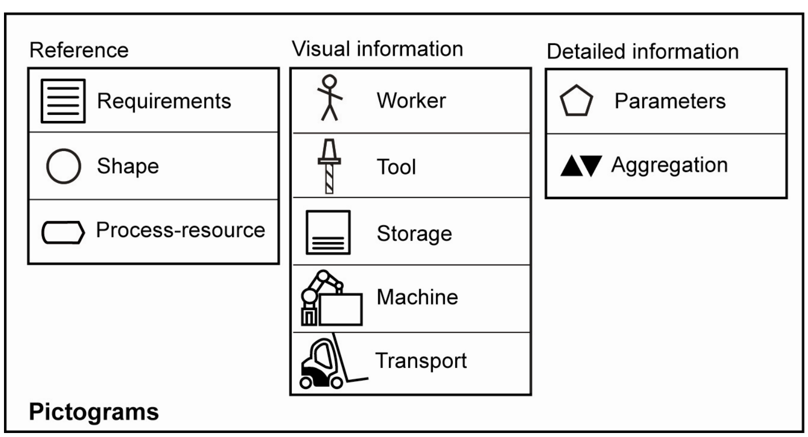

Figure 4-6: Types of pictograms

Pictograms used as a reference to partial models

References enable the representation of relationships between constructs and partial models. If a relationship between a construct and a partial model exists, the pictogram is placed. There are three pictograms indicating *references*: *shape, requirements* and *process-resources.*

Requirements pictogram: This pictogram makes reference to the *partial model requirements* (detailed information on Chapter 4.4.1). It is used on the *material element construct, software components construct, process construct* and the *resource construct.* It presents a list with the requirements that the construct fulfils and/or considers. Figure 4-7 presents an example of the considered requirements for the miniature robot's housing final machining process.

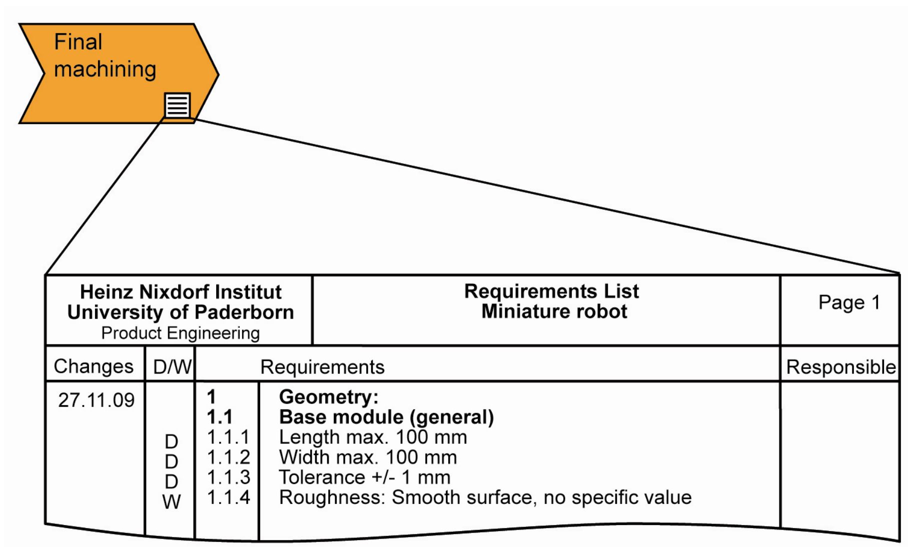

Figure 4-7: Example of the requirements pictogram

Shape pictogram: This pictogram makes reference to the *partial model shape* (detailed information on Chapter 4.4.4). It is used on the *material element construct* (shape of the product) and the *resource construct* (shape of the manufacturing system). A *shape pictogram* within a *resource construct* presents a shape related description of the resource in form of a CAD model, a sketch, graphic enhanced with text, etc. Figure 4-8 presents an example of the shape pictogram use. It presents a CAD model from the Robot Fanuc-S430i [3DC09-ol]. This robot is used for spot welding.

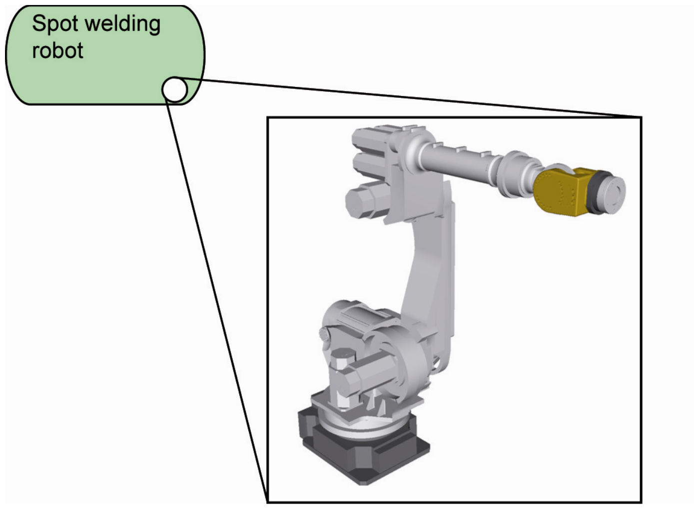

Figure 4-8: Example of the shape pictogram. Robot's picture taken from [3DC09-ol]

Process-resource pictogram: If the pictogram is placed on a process within the *partial model process sequence* then, the pictogram links the process with its corresponding resource (actuator). If the pictogram is placed on a resource within the *partial model resources,* then, the pictogram links the resource with its corresponding process (activity and/or operation that the resource fulfils). Figure 4-9 presents an example of an assembly robot (resource) and its corresponding task (process) from both perspectives.

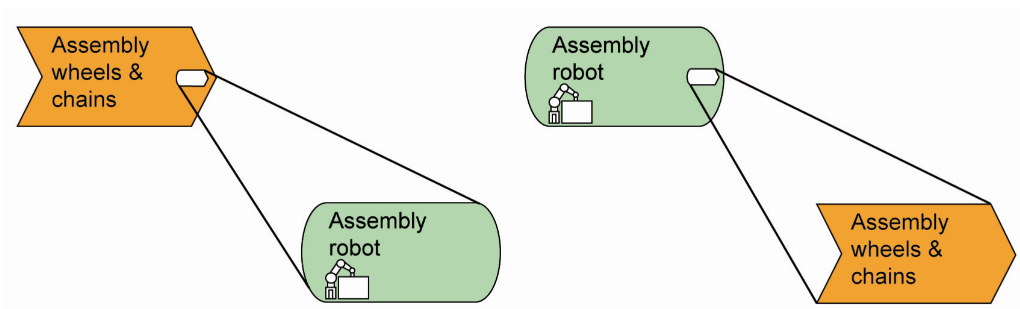

Figure 4-9: Example of a process linked to a resource and a resource linked to a process with the process-resource pictogram

Pictograms used as visual information

The pictograms in this group are used on the *resources construct* to communicate the user in a clearly and effective way the type of resource that is being represented. This is important because each type of resource implies different types of properties and characteristics. The use of the pictograms enables the user to estimate at a glance the capacity and attributes of each individual resource as well as its harmony within the production structure. There are five types of pictograms used to specify the type of resource: *Worker, tool, machine*, *storage* and *transport*. Figure 4-10 presents the mentioned pictograms.

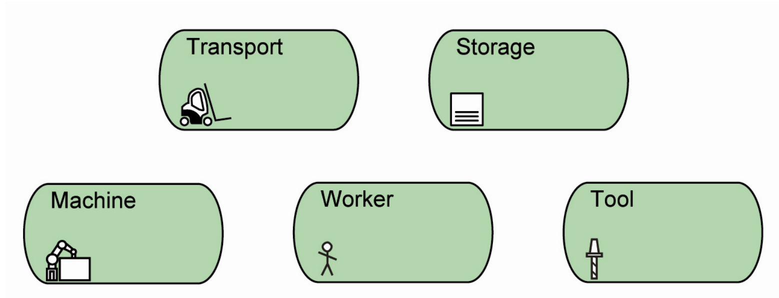

Figure 4-10: Pictograms used to specify the type of resource

Pictograms indicating detailed information

The third group of pictograms is used to detail information. There are two pictograms: the **parameters pictogram** and the **aggregation pictogram**.

Parameters pictogram: It is used to link the considered process or resource with a table where **relevant parameters and information** (characteristics of the process/resource) are collected.

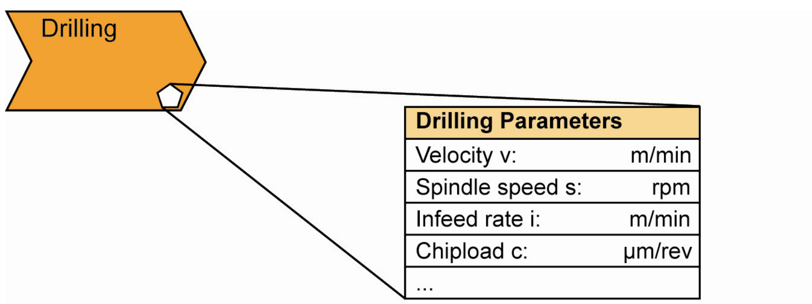

Figure 4-11: Pictograms used to specify parameters

According the Merriam-Webster dictionary "*a **parameter** is any of a set of physical properties whose values determine the characteristics or behaviour of something"* [Mer10-ol]. It is also a characteristic element. Relevant **process parameters** are the technic parameters of the process, for example, in Figure 4-11 relevant drilling parameters are presented, like the required spindle speed or the infeed rate. Relevant **resources parameters** include material, *structural information* of the resource[19], acquisition costs and variable costs, as well as technic parameters of the resource like achievable tolerance, production capacity, etc.

Typically, **process parameters** present **a concrete value**, the required value of a specific parameter. On the other hand, **resource parameters** present **the range of values** of a specific parameter that the resource can achieve. Relevant information and characteristics of a resource include, for example the *resource's structure*. If the resource has aggregated resources, in this field is possible to define which resources collaborate with

[19] *Structural information*: It describes the resource's components

the main resource by writing here its ID number. This theme is treated with more detail on Chapter 4.7.

Resources can be classified in five categories: *Worker, machine, tool, transport and storage.* Each category can have special relevant parameters, for example:

- *Storage:* Time, conditions (Temperature-controlled, etc.), load
- *Transport*: Load, distance, time
- *Worker:* Shift

Aggregation pictogram

Complex **processes** can be decomposed into its components. *Decompositions* allow the user to describe a process with different levels of abstraction. On the other hand, *aggregation* is the collecting of components into a whole. The decomposition principle is to divide. The aggregation principle is to join. Figure 4-12 presents an example of process' decomposition and aggregation.

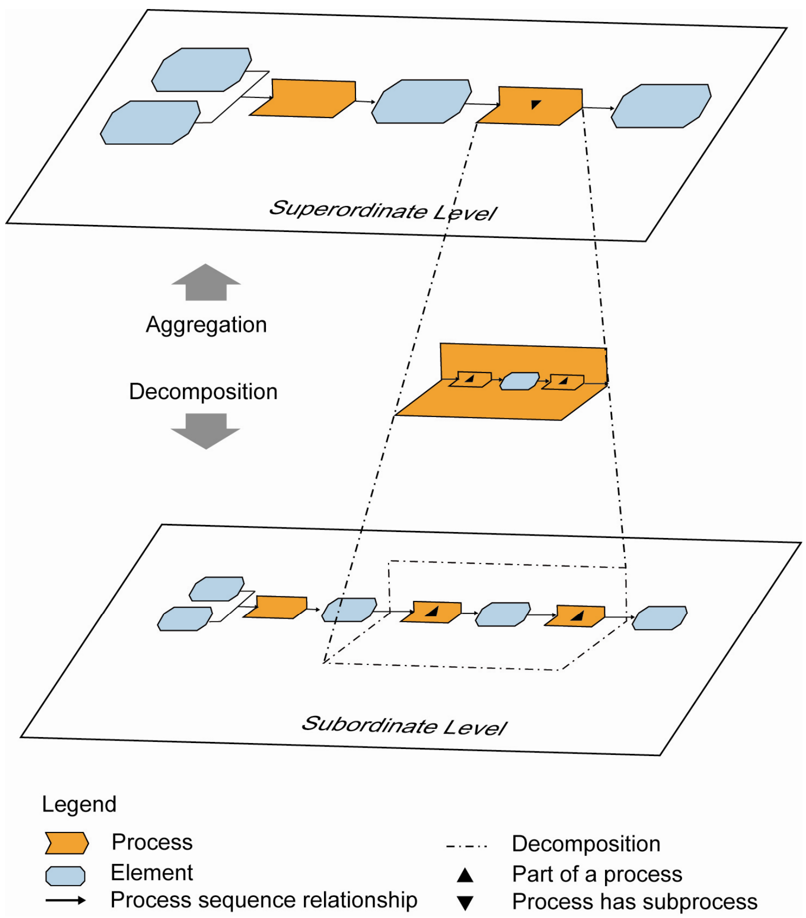

Figure 4-12: Example of a process decomposition and aggregation

Decomposition is represented with a black-pointing down triangle, the **aggregation pictogram**. The aggregation pictogram links the original process (superordinate level, Figure 4-12) with another sequence that represents an exploded description of the process, offering a higher level of detail about it. This new level (subordinate level, Figure 4-12) is known as the decomposition of the original process. Analogue to the original process, the subprocesses present also an aggregation pictogram, but inverse, a black-pointing up triangle. The subprocesses can be further decomposed, offering in each exploded description superior level of detail.

Figure 4-12 presents a process sequence on the superordinate level. The last process of the sequence is decomposed into two processes, giving to the subordinate sequence a higher level of detail. The same figure can be interpreted as an aggregation of the last two processes within the subordinate level into one process on the superordinate level.

Analogue to the process' aggregation, resources can be aggregated. Nevertheless, the aggregated resources can not have aggregated resources themselves. Resources are specific actuators of a technology. In some cases, resources work together to perform one single action, for example, a CNC machine with a milling cutter or a worker with a screwdriver. Figure 4-13 presents an example of this concept.

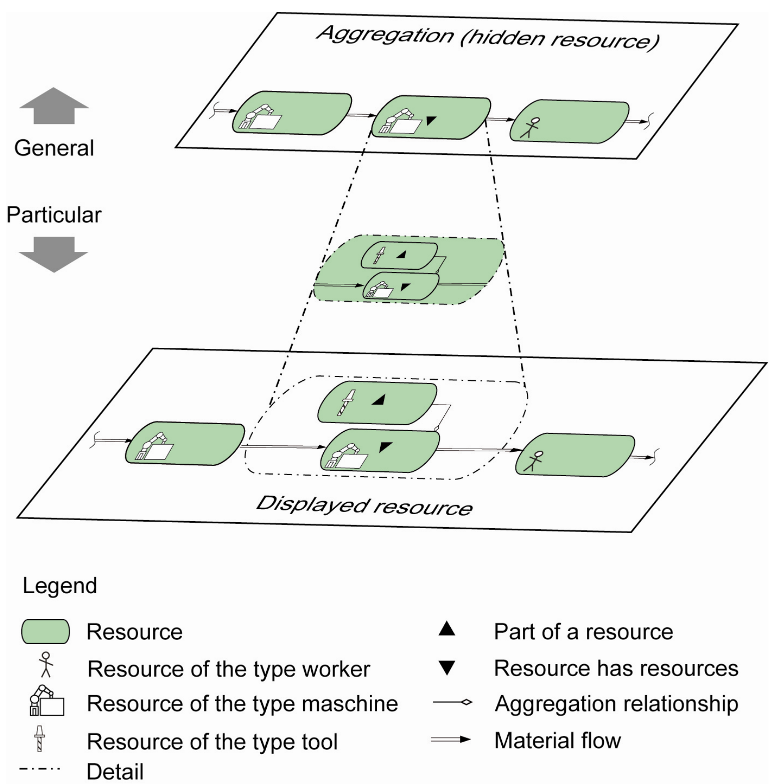

Figure 4-13: Example of aggregation of resources

In the upper diagram of the figure, the *machine construct* has an aggregated resource, which is hidden. If the user requires more detail it is possible to display the main resource with the hidden resources, as presented in the lower diagram of the figure. The aggregation of resources is depicted by means of an *aggregation pictogram* and an *aggregation relationship* for each aggregated resource (see Chapter 4.3.3).

4.3.3 Relationships

A relationship is a connection between model elements. There are three types of relationships: **sequence relationship, aggregation relationship** and **material flow**. Each relationship is represented with a different type of arrow.

A **sequence relationship** is a type of logic relationship that indicates a successional relationship. This relationship enables to link, alternately, **elements** and **processes**. The sequence relationship is depicted with a single arrow. Sequence relationships are unidirectional. Figure 4-14 illustrates a process sequence where the sequence relationship is displayed.

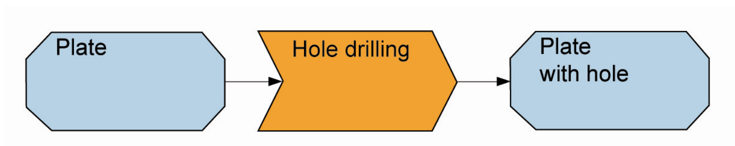

Figure 4-14: Graphic representation of a sequence relationship

An **aggregation relationship** is used to depict a **resources** object as **a part of**, or **as subordinate to another resource** object. Aggregation relationships are unidirectional and are used to increase the degree of detail of the model. They appear as a solid line with an unfilled diamond at the association end. The arrowhead points the main resource. The back of the arrow points to the aggregate part. An example of an aggregation relationship is depicted on Figure 4-15. The figure presents a *tool* that has an aggregation relationship with a *CNC machine*. The aggregation relationship is used here to indicate that the tool is part of the CNC machine. Aggregation is closely related to composition.

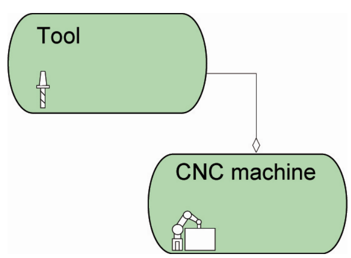

Figure 4-15: Graphic representation of an aggregation relationship

A **material flow** is a type of relationship that models the flow of elements from one **resource** to another one. **Material flow relationships** are unidirectional and appear as a solid double-line with a filled triangular arrowhead at the association end. As the Figure 4-16 illustrates, a material flow is displayed with a *material flow relationship* that goes from a *worker* resource to a *precision press* and then to a *worker*. Figure 4-16 depicts part of the production structure of a certain product. The worker manufactures the material and then passes it to the next station, the precision press machine. The machine processes the material and passes it to the next operation, which is executed by a worker. When the worker finishes, he or she passes the material again to next operation.

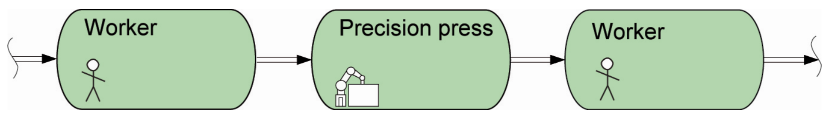

Figure 4-16: Graphic representation of material flows between resources

4.4 Partial models

Manufacturing systems are complex systems. For taming the system's complexity, the system was divided into aspects for its modelling. These aspects are: **requirements, process sequence, resources** and **shape of the manufacturing system**. To manage the information amount as well as its documentation, the manufacturing system's aspects are mapped on computer by **partial models**. These partial models were integrated to enable an holistic and integrative design. In the following paragraphs a description of the four partial models is presented.

4.4.1 Requirements

This partial model is part of the product's partial model requirements. This aspect considers the computer-internal representation of manufacturing systems' requirements. It presents an organised collection of requirements on the product and the manufacturing system. Shape related requirements (e.g. material, tolerances, surface finish etc.) as well as production related requirements (e.g. production volume, costs, deadlines) are of particular interest. They are used to derive requirements on the applied manufacturing equipment (e.g. the dimensions of the working spaces, speed of operation). The requirements are verbally described and concretized by attributes and their characteristics. There is a distinction between demands and wishes:

> *„Demands are requirements that must be met under all circumstances; in other words, if any of these requirements are not fulfilled the solution is unacceptable. Wishes are requirements that should be taken into consideration whenever possible, perhaps with the stipulation that they only warrant limited increases in cost, for example, central locking, less maintenance, etc." [PBF+07, Pg.147].*

Figure 4-17 illustrates a detail of a requirements list.

Heinz Nixdorf Institut University of Paderborn Product Engineering		Requirements List Miniature robot		Page 1
Changes	D/W	Requirements		Responsible
29.04.10		**1**	**Geometry:**	
		1.1	**Base module general**	
	D	1.1.1	Length max. 100 mm	
	D	1.1.2	Width max. 100 mm	
	W	1.1.3	Height, small	
	D	1.1.4	Optimal behaviour between the base area and the height	
	D	1.1.5	Low centre of gravitation	
	D	1.1.6	Modular construction	
	W	1.1.7	Robust	
	D	1.1.8	Elegant	
	W	1.1.9	Rounded form, circular	

Figure 4-17: Detail of the miniature robot's requirements list

4.4.2 Process sequence

The process sequence partial model is used to capture, manage, display and communicate process knowledge. This partial model **describes the manufacturing operations**[20] (activities) to produce a **single product** and **their sequence** as a chain of processes. In other words, it is used to model the chronological order of the processes required to manufacture a product.

Each process is characterised by a function and additional attributes (requirements, process parameter). Functions are described by a substantive and a verb (e.g. machining the housing, assemble wheels). Each process has at least one input-object and at least one output-object. The input and output objects are referred to as **elements** (see Chapter 4.3.1). Manufacturing processes carry out a transformation of components and assemblies with regard to form and material properties. To increase the clearness of the process sequence, it is possible to hide out the output objects of processes in which no transformation of the material element's form or properties occurs, for example with the process *positioning operation*. In Chapter 4.6.2 this theme will be treated with more detail.

Figure 4-18 illustrates a detail of the process sequence of the miniature robot's housing with drive mechanics. The module consists of MID-housing, electric motor, wheels and chains. The manufacturing of the MID-housing is shown in the upper sequence. The initial process is the master forming of the housing by injection moulding. Thereby plastic granules are transformed into an unprocessed housing. The process injection moulding is characterised by requirements and process parameters.

Examples for requirements are allowed materials, draft angle and maximum thickness. Pressure, moulding holding time and cooling time are only some of the examples of process parameters. The final machining closes the housing's manufacturing, the result

[20] With the term *manufacturing operations* is referred to manufacturing, assembly and logistic processes. Please refer to Chapter 4.3.1 for more information.

is the finished MID-housing. Requirements and shape of the MID-housing are specified. The usage of liquid crystal vectra is demanded, a dark colour is wished. The shape is stored as written specifications, e.g. dimensions and tolerances.

The lower sequence shows the manufacturing of the drive wheel. This process is very abstract because the available information is incomplete. Only input and output objects are known. The assembly is carried out in three steps. Firstly, the wheel and the bearing are assembled. The result is the drive wheel. Secondly, the electric motor is mounted into the MID-housing. Those two assembly processes are independent from each other. The final process is the assembly of the completed housing, including the electric motor, carrying and drive wheels and track chains as input. The produced output is the housing with drive mechanics.

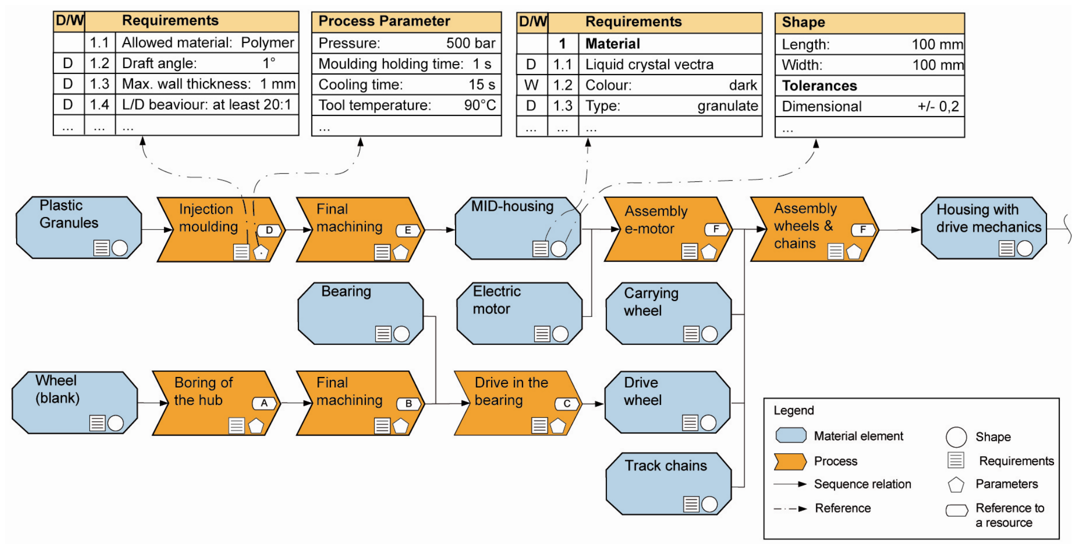

Figure 4-18: Process sequence of the miniature robot BeBot (detail)

4.4.3 Resources

The partial model process sequence is designed to plan the manufacturing process of a single product, while the **partial model resources** is designed for planning the required amount of products over a period of time, in other words, the required production. Thus, the resources partial model takes into account resources, production rate and material flows of the system. For example, the process *assembly wheels & chains* (F) in Figure 4-18 can be easily made by different resources. If the desired production rate is one product each five minutes, a resource w*orker* may be enough to accomplish the job. But, if the required production rate is one product per minute, then, the resource *assembly workstation* may be more suitable (see Figure 4-19).

Analogue to the partial model process sequence, the partial model resources is depicted graphically by means of a diagram. This diagram is used to represent knowledge about the production structure and resources that participate in the product's manufacture. It is also used to capture, manage, display and communicate information about the material flow through the process actuators.

The diagram is a model of the resources sequence, which not always is the same as the process sequence. There are many reasons that explain this situation. Resources can execute more than one process, implying loops within the resources sequence. Resources can work in parallel, executing two different processes on the same product. Resources can produce a required attribute (e.g. a hole), in many exemplars of a product (e.g. three exemplars of a product can be manufactured at the same time within a single machine).

The resources are connected by arrows indicating the material flow. The material flow is derived from the process sequence. Resources are concretized by requirements, parameters and shape. The resources related to the process sequence of the miniature robot (Figure 4-18) are illustrated in Figure 4-19. The manufacturing of the housing is shown in the upper sequence. The initial resource is the injection moulding machine. It executes the process injection moulding. The required production rate is 12 pieces per hour. The resources parameters presents a range of parameters and values that the resource is capable to perform. In this example details about pressures are particularly important, for example the possible boost is between 500 and 1500 psi. The injection moulding machine requires an area of 2000 x 5000 mm with a maximum height of 2600 mm. The three resources in the lower sequence represent the manufacturing of the drive wheel. In the course of the concretization, the resources are detailed by further resources, e.g. machines by tools. In Figure 4-19 the CNC lathe is detailed by the lathe tool. This is done because information about the used lathe tool is of prime importance for the interaction between product and manufacturing system.

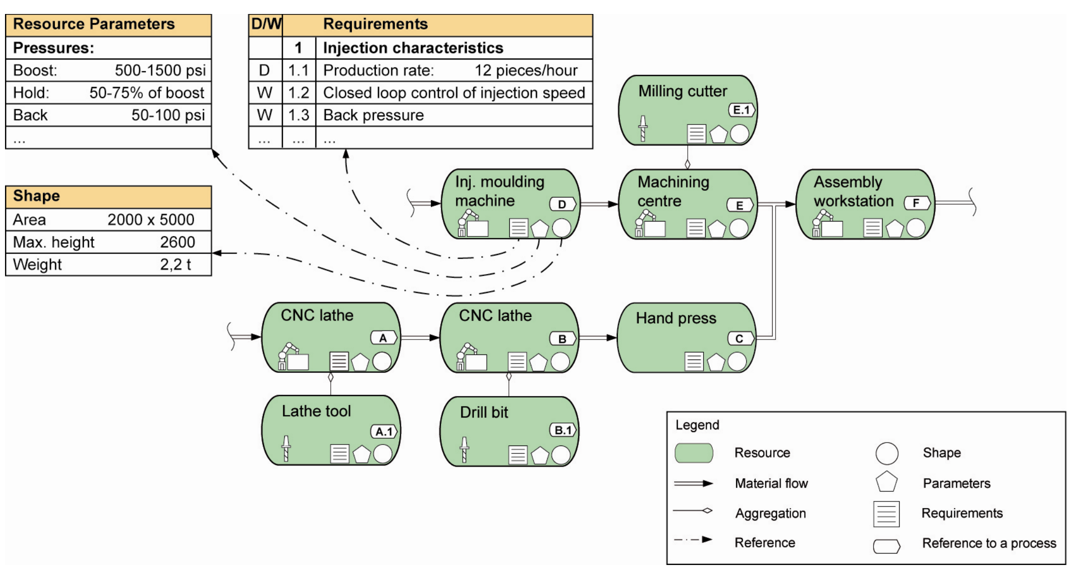

Figure 4-19: Depiction of the partial model resources of the miniature robot BeBot's manufacturing system (detail)

4.4.4 Shape

Analogue to the conceptual design of the product, first definitions of the shape are made during the conceptual design of the manufacturing system. With shape is meant the workspace, the required floor space of machines or the active areas of handling appliances. The information is stored as written specifications, sketches or CAD data. Information regarding the shape of the manufacturing system is necessary for the concretization within the development process especially for the place of work planning and the working appliance planning.

Figure 4-19 illustrates a detail of the resources needed for the robot BeBot's manufacture. The resource's shape, in this case the shape of the injection moulding machine, is referred to with the shape pictogram. The machine requires an area of 2000 x 5000 mm with a maximum height of 2600 mm.

4.5 Specification of the partial models' cross-linking

The partial models and their interaction represent and document the product's manufacture in the course of the system's concretization. Each partial model represents *a closed structured group of individual objects* which interact with other objects of other partial models. These objects are independent objects. To represent these connections, references are used. A model of a system is thus a network of interrelated objects. Figure 4-20, 4-21 and 4-22 present a synoptic view of the possible relations between the different objects of the system.

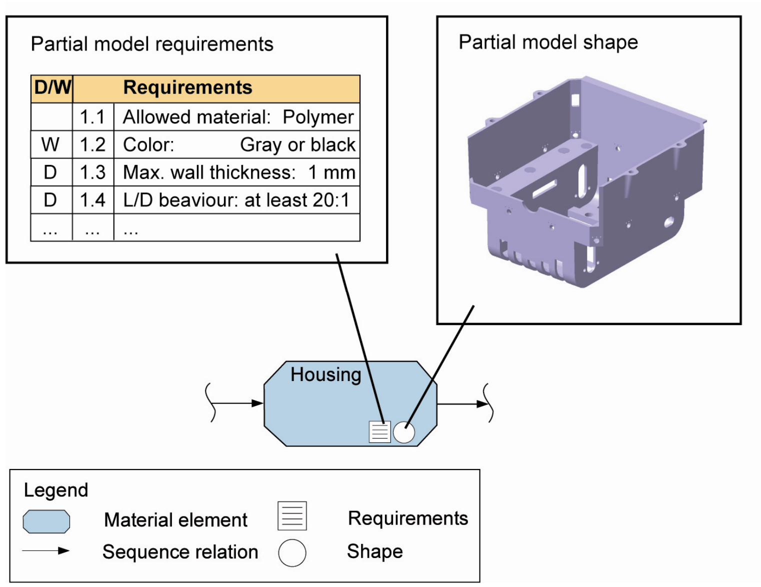

Figure 4-20: Relationships between the element construct and other objects within the specification technique

Figure 4-20 presents the possible connections of the **element** construct with other elements of the specification technique. In this case, the element *housing* is linked with two partial models, the partial model requirements and the partial model shape (of the product).

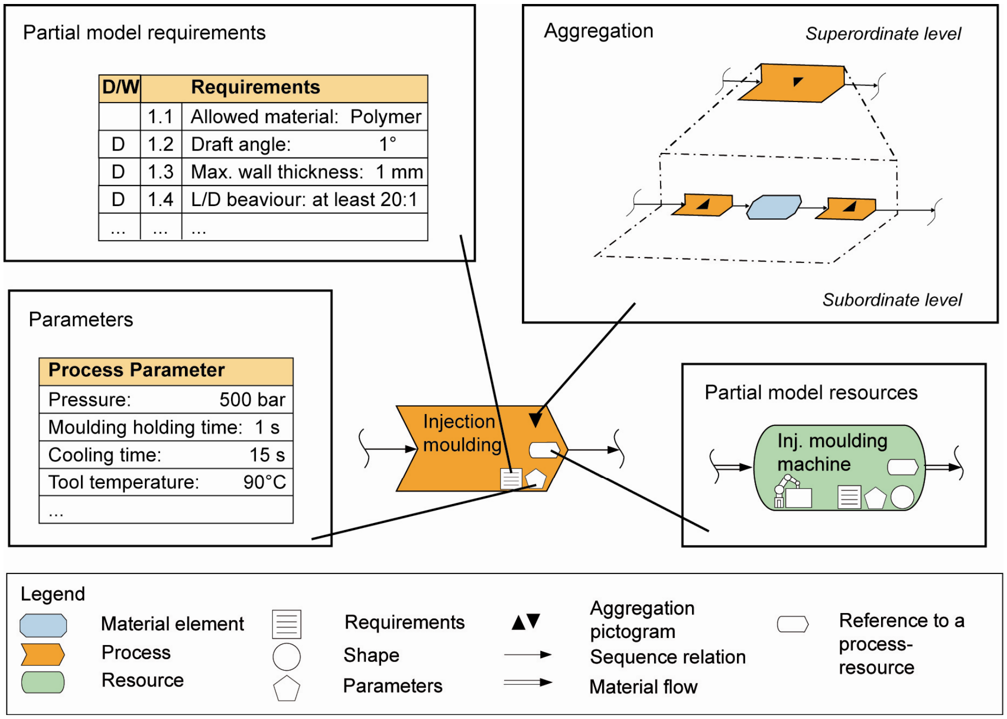

Figure 4-21: Relationships between the process construct and other objects within the specification technique

Figure 4-21 presents the possible connections of the **process** construct with other elements of the specification technique. In this case the process *injection moulding* is linked with two partial models, the partial model requirements and the partial model resources. It is also linked with a table where relevant parameters and information are collected *(parameters pictogram)*. Finally the process is linked with another sequence that represents an exploded description of it (*aggregation pictogram)*.

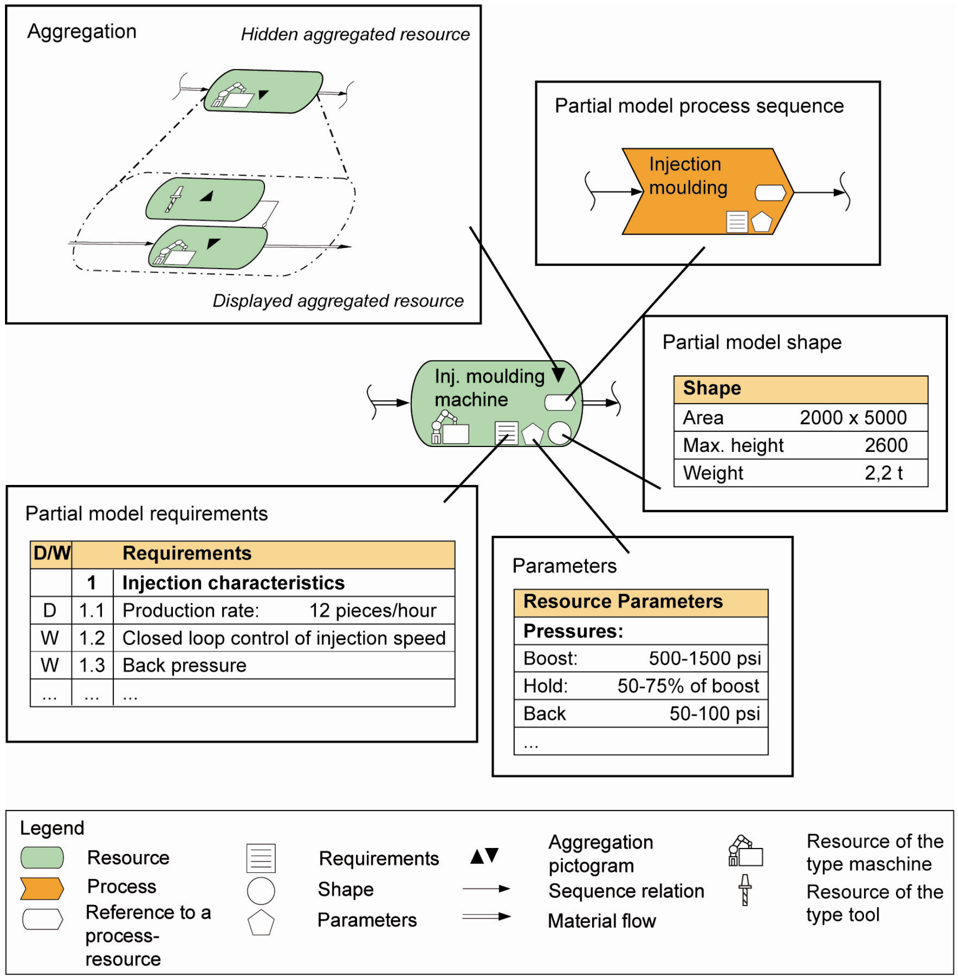

Figure 4-22: Relationships between the resource construct and other objects within the specification technique

Figure 4-22 presents the possible connections of the **resource** with other elements of the specification technique. In this case, the resource *injection moulding machine* is linked with three partial models, the partial model requirements, the partial model process sequence and the partial model shape. It is also linked with the table where relevant parameters and information are collected (*parameters pictogram)*. This resource has an aggregated resource. This linkage is represented with the *aggregation pictogram* (see Chapter 4.3.2).

The above mentioned interrelations are going to be explained using the example of the miniature robot's drive wheel, depicted in Figure 4-23. The figure presents a detail of the miniature robot's manufacturing system. Among the requirements collected within the *requirement list*, we can find the drive wheel requirements as well as requirements on the manufacturing system and its resources (1).

In order to produce the wheel, suitable processes and their sequence are selected. In this case, a *boring operation*, a *final machining* and an assembly operation (*drive in the bearing*) are needed to produce the wheel. Each process and each element satisfy at least one requirement contained within the requirements list. For this reason, *each element* and *process construct* has a *requirements pictogram* which makes reference to this dependence. Aside from that, *element constructs* contain a link to the partial model *shape of the product's*. This link is depicted by the *shape pictogram*. Once that the process chain is defined, resources are allocated to the processes (2) & (3). The linkage is represented as a *reference pictogram between process - resource*. In this case, the *final machining* and the *drive in the bearing* are allocated to two different resources, a *CNC lathe* and a *hand press*. Finally the *resource construct* is linked with detailed information of the analysed resource, in this case, a CAD model with the CNC lathe's spatial information (4). This linkage is represented as a shape pictogram.

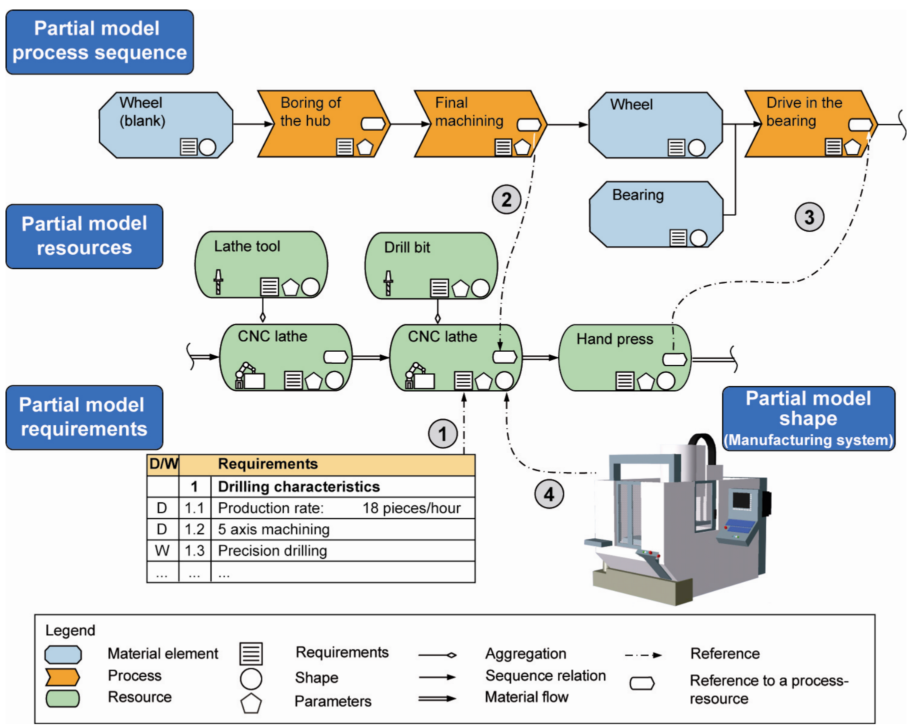

Figure 4-23: Depiction of the partial models cross-linking

4.6 Modelling rules

Modelling rules specify how the modelling objects are to be used as well as their meaning. Modelling rules help to improve the quality of a model by improving its clearness and readability. Furthermore, modelling rules help to avoid misinterpretations and makes a model easier to maintain as well as enhance. This chapter presents the specification technique's modelling rules.

4.6.1 Basic rules

Modelling the requirements

The requirements are collected within a list. At the beginning of the product conception, the requirements are abstract. These contain the desired functionalities and properties of the product. In the course of the product and manufacturing system concretization, the product's *properties*[21] and functionalities are translated into concrete manufacturable components and *characteristics*[22]. The requirements list is the record of this concretization process. It is a continually reviewed document where the last changes and demands are reflected. It serves as an up-to-date working document. Figure 4-24 presents the typical information and layout of a requirements list.

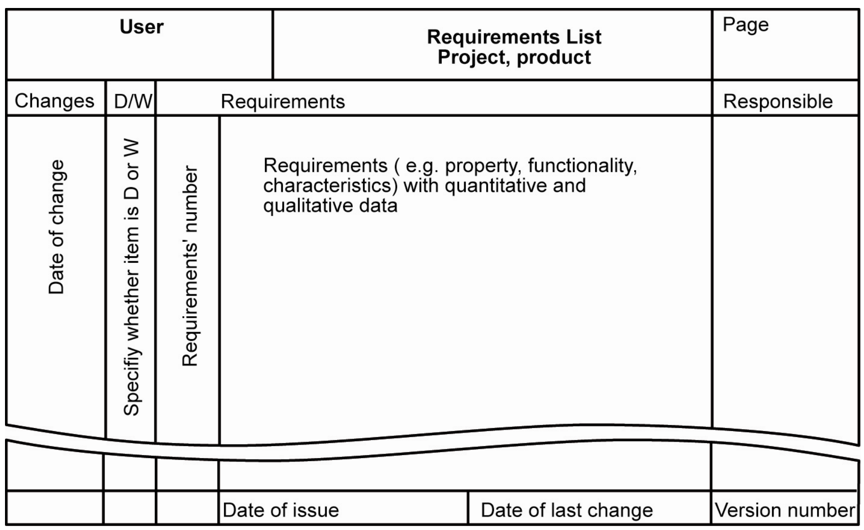

Figure 4-24: Layout of a requirements list (adapted from [PBF+07, Pg.148])

Modelling the process sequence

The process sequence partial model is depicted graphically by means of a diagram. In this section, some examples are given, illustrating how the constructs are used. The syntax for the simplest case—a single material element attached to a process in a basic transformation operation—is illustrated in Figure 4-25.

---

[21] *"The* ***properties*** *describe the product's behaviour (e.g. weight, safety and reliability, aesthetic properties, but also things like "manufacturability", "assemblability", "testability", "environmental friendliness", cost). They cannot be directly influenced by the developer/designer" [Web07, Pg. 87]*

[22] *"The* ***characteristics*** *describe the structure, shape, dimensions, materials and surfaces of a product ("Struktur und Gestalt", "Beschaffenheit") These can be directly influenced or determined by the development engineer/designer" [Web07, Pg. 87]*

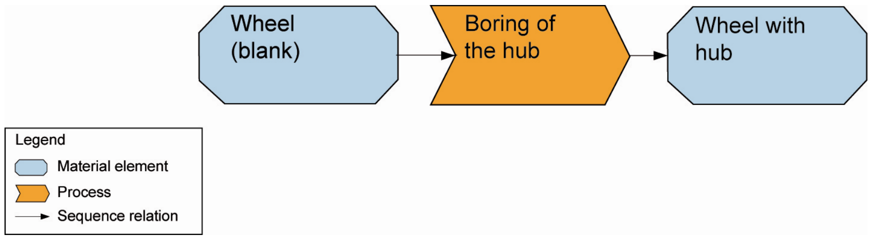

Figure 4-25: Depiction of a basic transformation operation

In general, the semantic of a basic process is that, in an occurrence of the indicated process, there is first an object *A* in state *a*, which is transformed/changed by means of the process P. As a consequence of the transformation, the object *A* comes to be in state *b*; it is *permitted*, though perhaps not typical, that the object in state *a* be distinct from the object that comes to be in state *b* (e.g. separation processes, where one object becomes two or more objects); and it is *permitted*, though perhaps not typical, that *A* remain in state *a* after the process (e.g. storage process). Figure 4-25 illustrates a basic process (the operation “boring”) involved in the transformation from the material element “Wheel” to “Wheel with hub”. Thus, the general semantic of the basic process *boring* is that, there is first an object *wheel*, which is transformed by means of the process *boring*. As a consequence of the transformation, the object *wheel* comes to be the object *wheel with hub.*

Manufacturing operations commonly involve the following transformations, transitions and changes: Geometry and /or properties transformation, change of state (e.g. temperature change), position (e.g. displacement), to place or leave in a location for preservation or later use, separation processes and assembly operations. Figure 4-26 present some examples of these operations. Additionally it presents an example of a process sequence diagram. It illustrates the main assembly process used on the manufacture of the miniature robot.

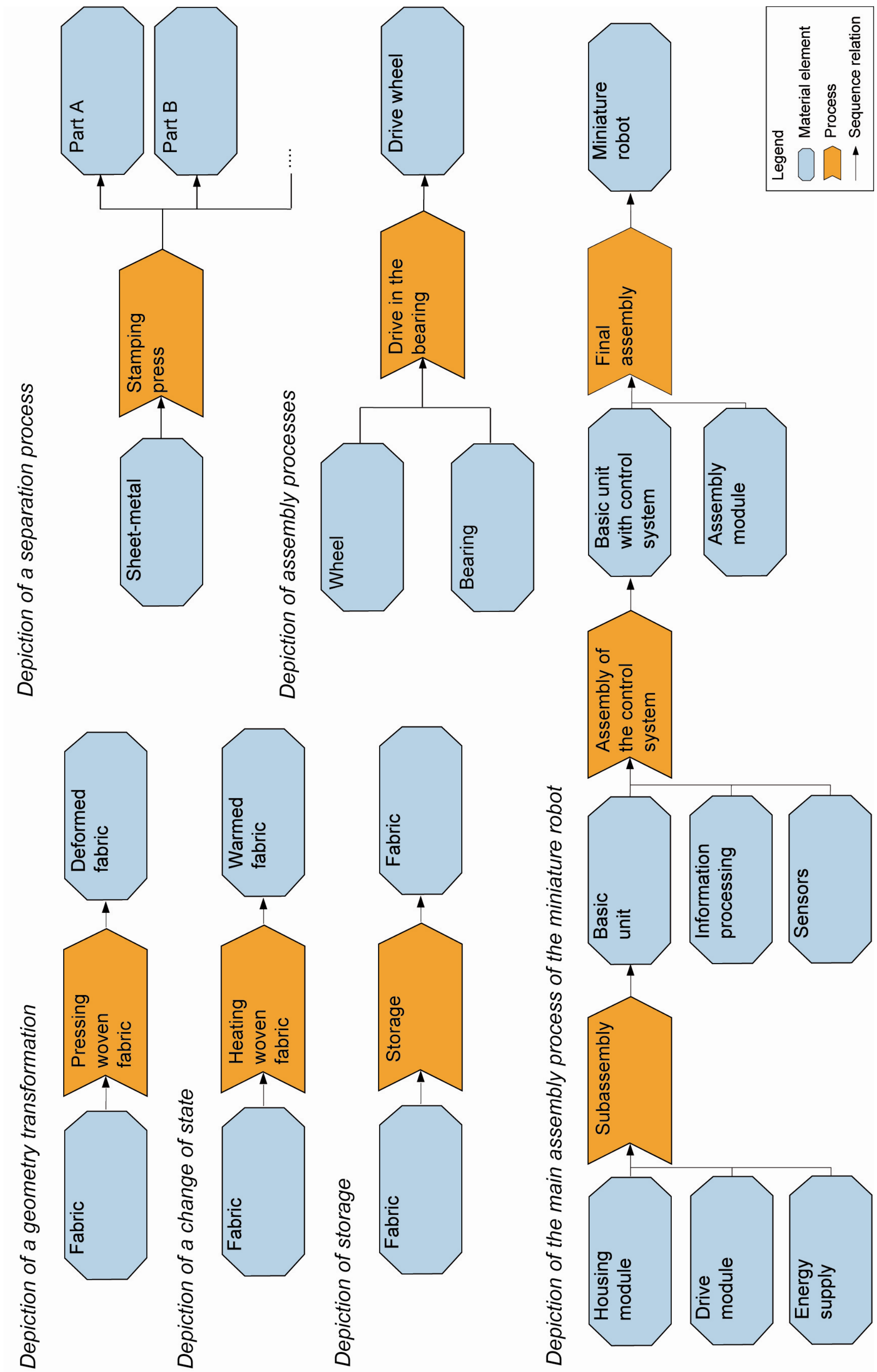

Figure 4-26: Depiction of common manufacturing operations

Modelling the resources

The resources partial model is depicted graphically by means of a diagram. In this section, some examples are given, illustrating how the constructs are used to develop the diagram. The syntax for the most typical case, a single transformation, is illustrated in Figure 4-27.

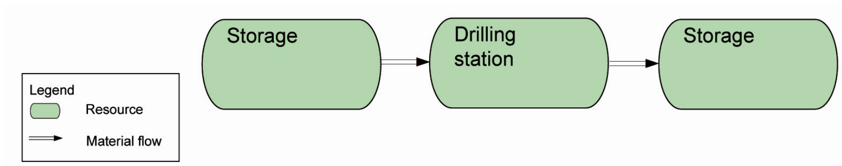

Figure 4-27: Depiction of a simple resources diagram

In general, the semantic of a basic *resources diagram* is that, firstly, there are X number of objects stored on the *storage* resource. Then, the objects are transferred to the next resource. This transfer is depicted by means of the *material flow* arrow. After that, the objects are transformed in this resource. Once that the operation is performed, the objects are transferred to the next resource, in this case a *storage* resource. Figure 4-27 illustrates this concept using as an example a simple material flow through a single resource, a drilling station. Thus, the general semantic of the diagram is that, there are objects stocked on the *storage resource*. These objects are transferred to the *drilling station* and then to the *storage resource.* Typically a resource sequences involve the following possibilities: Material flow through single resources, confluence from two material flows to one resource and a split material flow. Figure 4-28 presents illustrations of the mentioned examples.

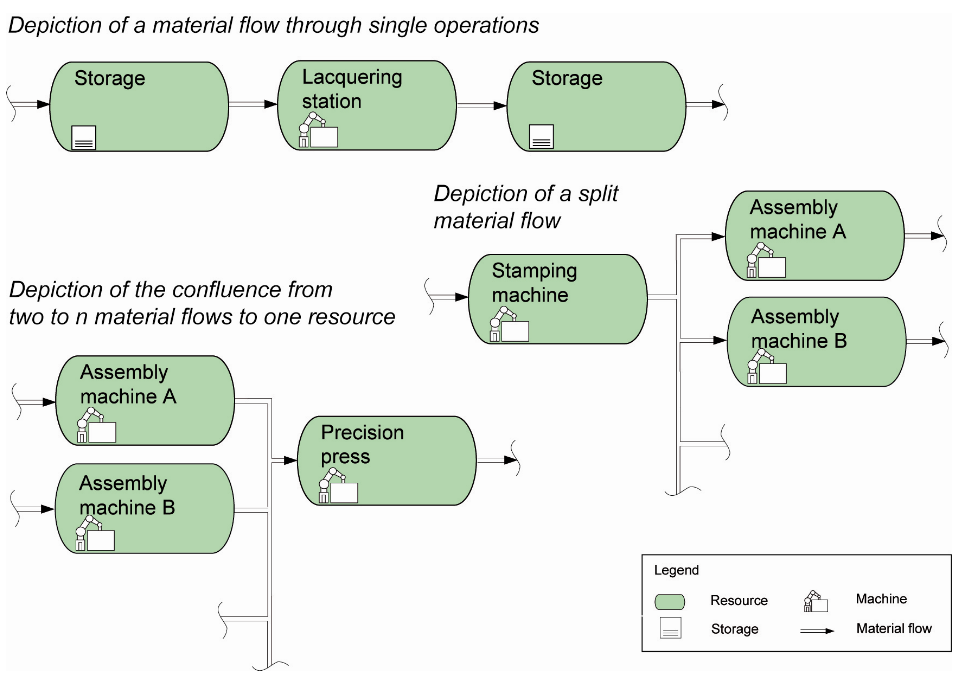

Figure 4-28: Depiction of common resources sequences within a resources diagram

Modelling the shape

Contrary to the partial model process sequence and the partial model resources, the partial model shape of a manufacturing system has no own diagram. The shape views are depicted and/or stored as written specifications, sketches or CAD data. Figure 4-29 illustrates a resource, a spot welding robot. The figure presents a CAD model of the robot [3DC09a-ol], as well as sketch of its working range [ABB10-ol]. Within the written specifications, relevant characteristics of the robot can be found. For example, the machine requires an area of 1107 x 720 mm and has a maximum weight of 1310 kg.

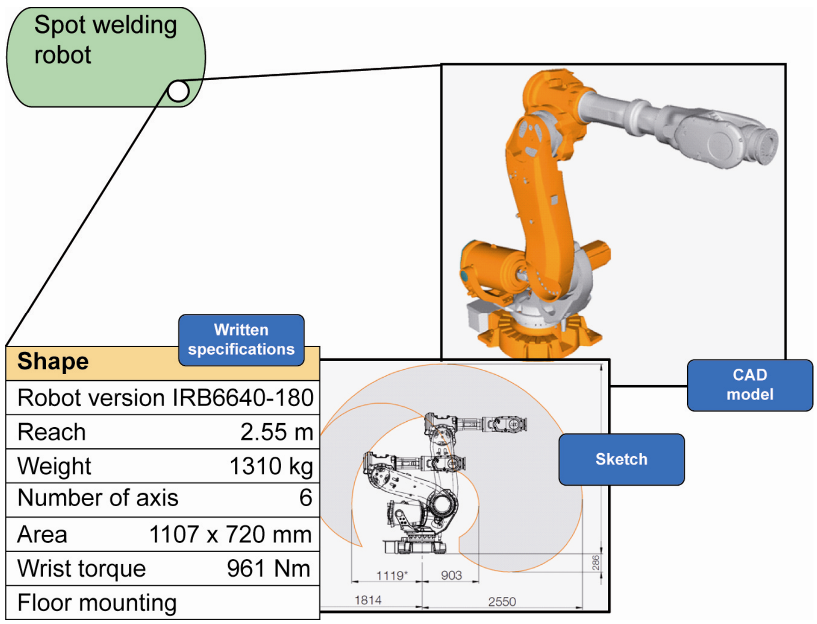

Figure 4-29: Depiction of the resources view, sketch taken from [ABB10-ol], CAD model taken from [3DC09a-ol]

4.6.2 Modelling special cases

Models are simplified depictions of the reality. They allow different representations of a same reality. In *special cases*, the developers may require to hide out some objects or views of a model to increase it comprehensiveness. For this reason, in the following section, special cases are going to be modelled and explained.

Hiding out constructs

Each process has at least one input **element** object and at least one output **element** (see Chapter 4.4.2). Nevertheless, to increase the clearness of the process sequence, it is possible to hide out the output objects of processes in which no transformation of the material element's form or properties occurs, for example, the process *storage*. It is also possible to hide the output objects of processes which are not "relevant" for the developer. With "hiding out" is meant that a construct is not depicted, at least not on the analysed view/partial model. With construct it is meant any element, process and/or resource. The construct may exist, nevertheless due to the selected approach or the abstraction degree, it may not be relevant. Nevertheless it can be considered on further views.

Figure 4-30 presents an example where the output of the “boring process” and the “final machining” are hidden out. Although the material element is not depicted, the general semantic of the depicted process sequence is that, there is first an object *wheel*, which is transformed, firstly, by means of the process *boring of the hub*, and afterwards by means of the process *final machining.* The output object of these processes and the *bearing*, are transformed by the process *drive in the bearing*. As a consequence of the transformation, the original object *wheel* comes to be the object *drive wheel.*

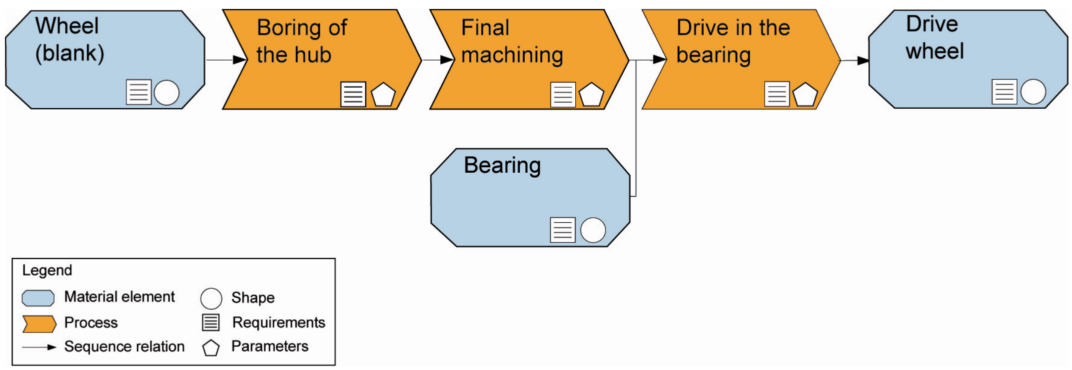

Figure 4-30: Depiction of a sequence operation with hidden out elements

Modelling decompositions of processes

If a **process** is highly complex, or the abstraction degree is really high, then it is useful to decompose the process into its components. A decomposition is represented with the **aggregation pictogram** (see Chapter 4.3.2). The aggregation pictogram links the original process with another sequence that represents a decomposition of the process, offering a higher level of detail about it. Processes within this new level, the decomposition of the original process, can have also a decomposition of themselves. In this way it is possible to structure a process description to any level of detail. Analogue to the original process, the subprocesses present also an aggregation pictogram, but inverse.

Figure 4-31 presents an example of a decomposed process sequence. The figure depicts three levels of a process. To manufacture the *End product,* three elements are required and a *Process 1* (Level U). Nevertheless, *Process 1* is not a single process. It consists of three subprocesses: *Process 1.1, Process 1.2* and *Process 1.3*. The figure shows how the three original input elements of the *Process 1* are reassigned to different subprocesses (Level U+1). The *Element A* and *Element B* are the input elements of the *Process 1.1. Element C* is the input element of the *Process 1.2*. Level U+2 presents the decomposition of *Process 1.1* into two subprocesses *Process 1.1.1* and *Process 1.1.2.*

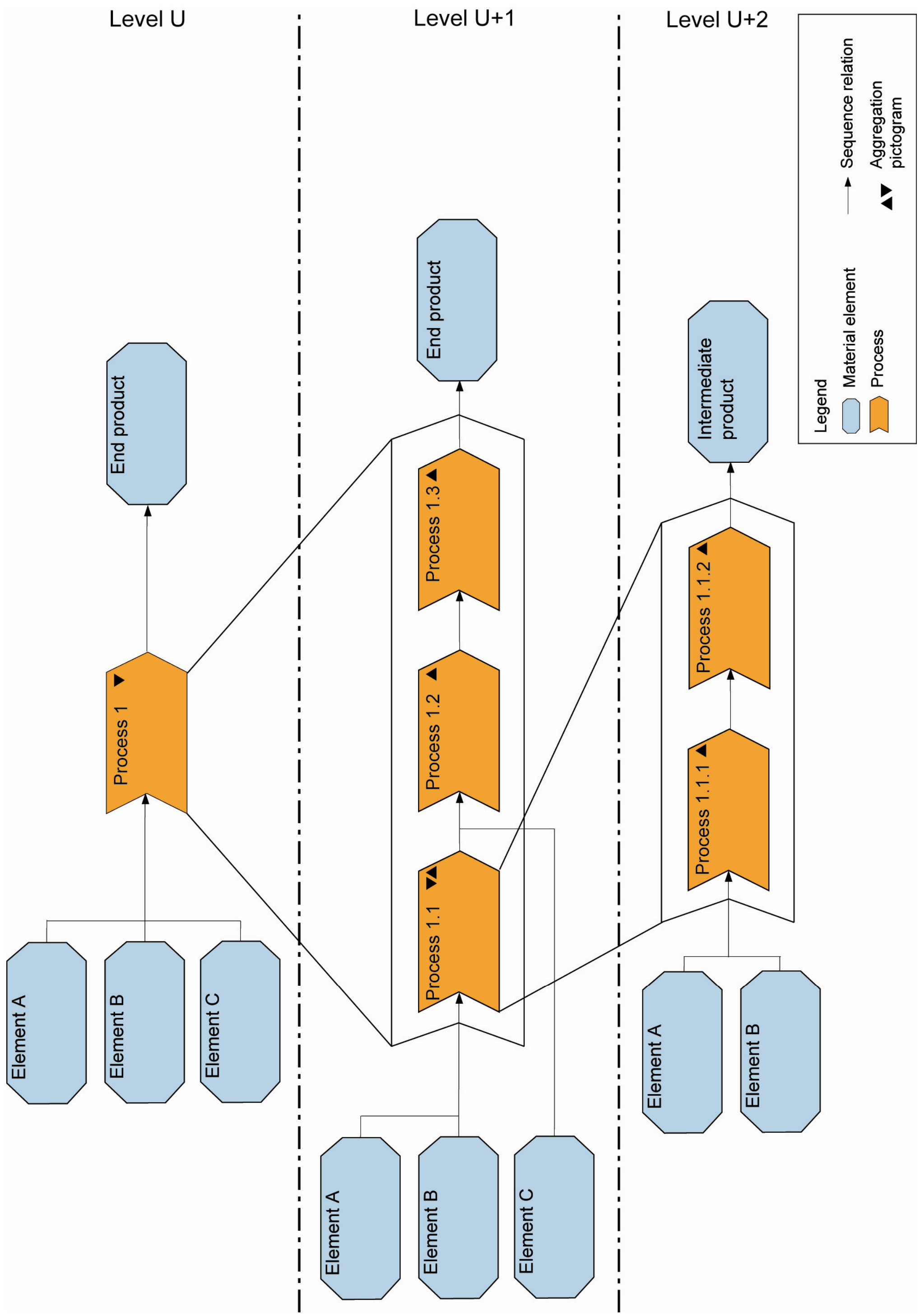

Figure 4-31: Example of a process decomposition

Modelling aggregations of resources

In this work**, resources** are considered **specific actuators** of a technology (e.g. Technology: *Vertical machining centre*. Resource: *Chiron FZ08 vertical machining centre*).

In some cases, resources work together to perform an action, for example, a machine with a tool or a worker with a machine. For this reason, resources can be aggregated. Nevertheless, the aggregated resources can not have aggregated resources themselves. This is because this diagram shows the **material flow between resources**. The **material flows within a resource** (e.g. the handling operations that a worker does) or **within a resource that cooperates with other resources** (e.g. within a CNC machine with many tools) are not considered.

Therefore, each *resource* with an *aggregated resource* can be depicted in two ways: the *hidden* form, where the *aggregated resource* is hidden and the *displayed* form, where the *aggregated resource* is displayed. Figure 4-32 presents an example of both depictions. The upper diagram has a resource, *a CNC machine*, with an **aggregation** pictogram. In this diagram, the *aggregated resources* are hidden. The lower diagram of the figure presents the same construct, but with its *aggregated resources* displayed. In this case, the diagram shows that the *CNC machine* resource has three aggregated resources: *milling tool, drilling tool* and *engraving tool*. Each of the aggregated resources has an *aggregation pictogram*, indicating that the resource is aggregated to a main resource.

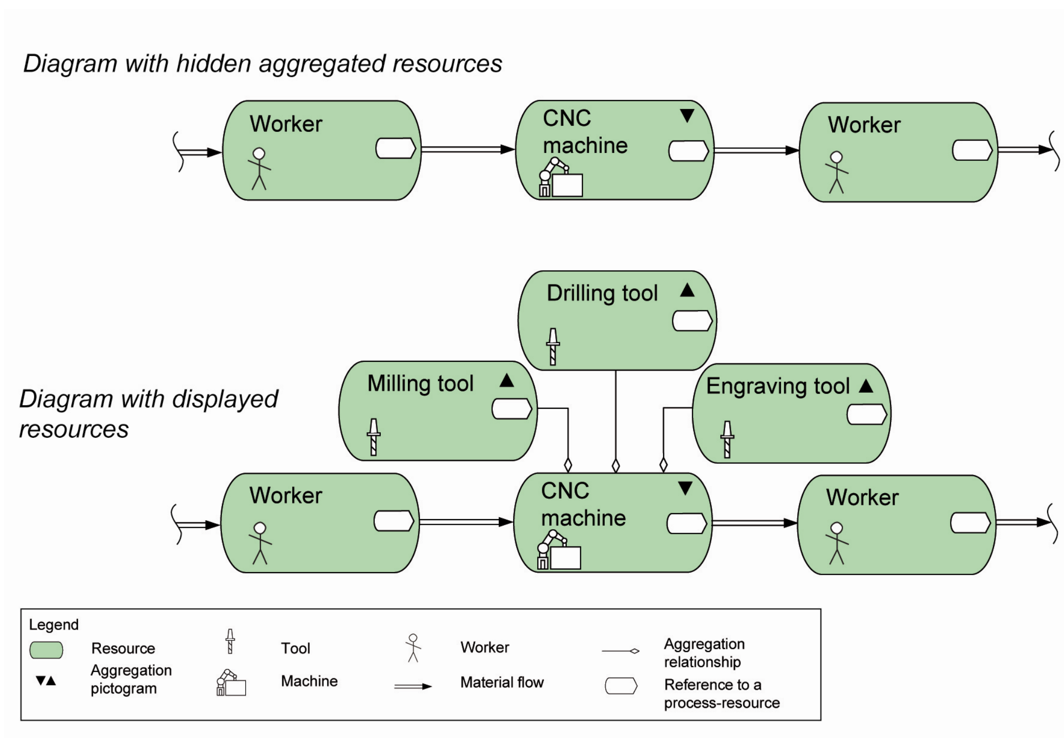

Figure 4-32: Detail of a diagram with resources aggregation

A close inspection of the resources' sequence makes quite clear that the process planning cannot be deduced from the resources diagram. This is because the material flow goes through *main resources* (in this case the *CNC machine*) and not through *the aggregated resources* (*milling tool, drilling tool* and *engraving tool*). It is not clear if the first operation is made by the milling tool, the drilling tool, the engraving tool or all these tools are used at the same time. The user of this specification may assume that the first aggregated process, the milling tool, may be the actuator of the first operation and the drilling tool may be the actuator of the second operation and so on. This assumption may be correct or not. The only way to know the process sequence is to check the process sequence within the process sequence diagram.

Modelling references between aggregations and decompositions

In the following paragraphs a special type of connection, the relationship between *decomposition of processes* and *aggregation of resources* is going to be explained. The **partial model process sequence** and the **partial model resources** are linked with a reference, the process-resource pictogram (see Chapter 4.3.2). This pictogram is placed on the resource construct and on the process construct. This relationship is used to detail the connection between process and resource.

The process sequence diagram can have many levels of detail (decomposition), while the resources diagram for its part not (see Chapter 4.3.2). Each process has one resource assigned and each resource executes one or more processes. Figure 4-33 illustrates the case of a *CNC machine* which executes *machining processes*. Looking with more detail, we see that the *CNC machine* has three aggregated resources, each one of them executes a specific task: *milling, drilling* and *engraving*. Each process has its counterpart on the resources diagram.

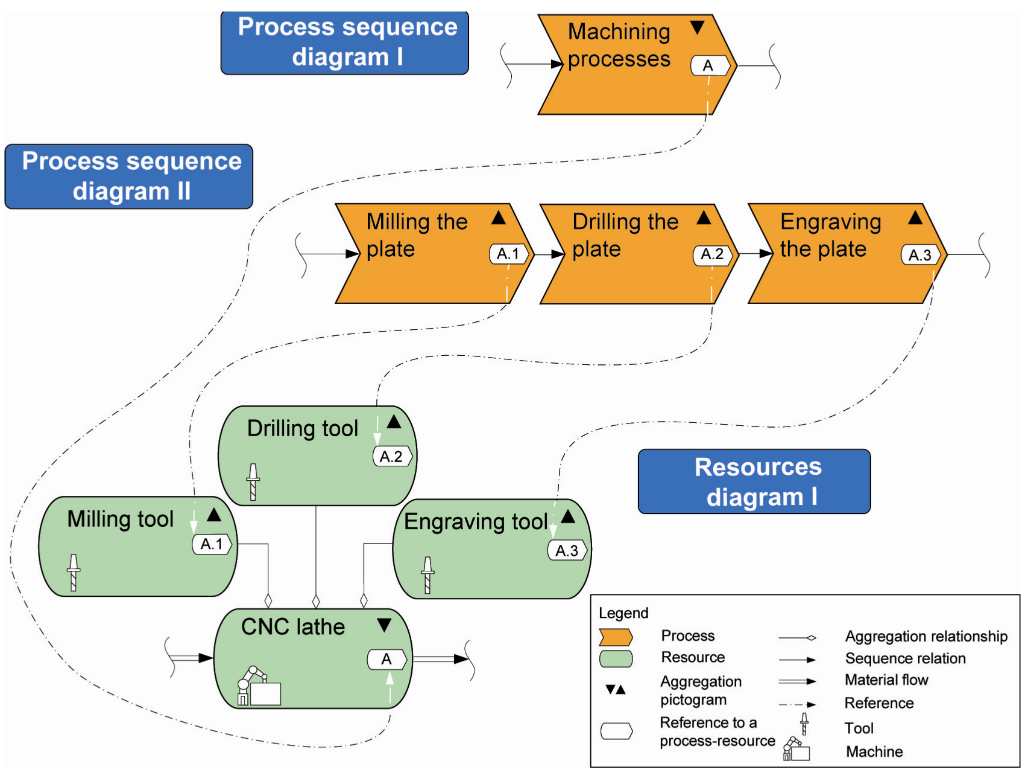

Figure 4-33: Example of a process decomposition linked to its resources diagram

A process can be decomposed in simpler subprocesses. In its turn, each subprocess can be further decomposed. Each process decomposition is depicted by process sequence diagrams. Nevertheless, the resources diagram works different. The resources diagram uses the principle of **detailing** and not decomposition. This implies a **dependency** between the main resource and the aggregated resources. For this reason each time an aggregated resource is referred, its main resource is also referred. Using the Figure 4-33 as example, the *process sequence diagram I* has a *machining process*. A decomposition of this process is depicted within *the process sequence diagram II*. The single process that compose the *machining process* are: *milling the plate, drilling the plate* and *engraving the plate*.

If the user wants to see the resource that performs the *engraving of the plate*, the user is going to see that the process *engraving plate* is made by the resource *engraving tool*, and that the *engraving tool* is an aggregated resource from the *CNC machine* resource. In other words, the user is never going to see the *aggregated resources* alone, because the *aggregated resources* are always depicted with its main resources. The *aggregated resources* are part of the *resources diagram I* and may be hidden or displayed.

Summarising, each *process decomposition* is depicted in a new process sequence diagram, which describes the original process in greater detail. *Aggregated resources* are dependent objects that need to be depicted **always** with its *main resource* within the *resources diagram*.

Modelling loops within the resources diagram

Resources can execute more than one process, implying loops within the resources sequence. Figure 4-34 presents an example of such a case, depicted in two different ways. The reading direction goes from left to right and from top to bottom. Each material flow involved within the loop's modelling is marked with a number. The first sequence of the figure illustrates a detail of a resources diagram in which the *material flow* (material flow 0) goes to a *CNC machine*. Here the material is worked and sent (material flow 1) to *worker X. Worker X* processes the material and sends it back (material flow 2) to the *CNC machine*. The *CNC machine* works up the material and despatch it (material flow 3) to *worker Y*.

The second sequence of the figure presents the same resources sequence's detail but with unmarked material flows. The reading direction, from left to right and from top to bottom, remains. The numbers depicted on the first diagram are used as an aid for the diagram's reading, nevertheless its use is optional.

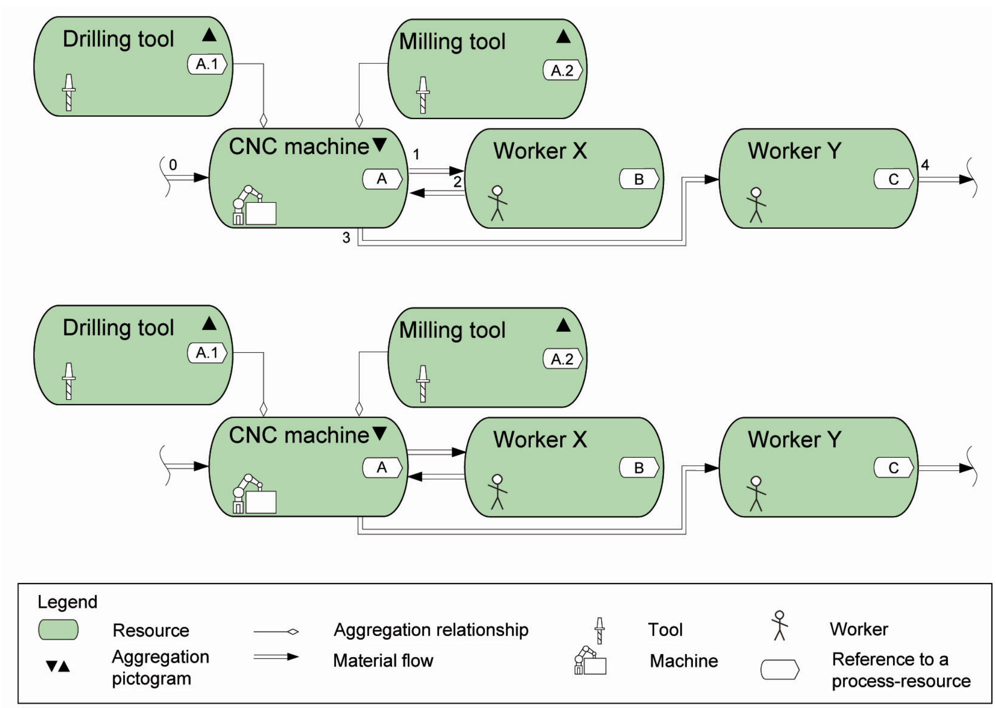

Figure 4-34: Example of a loop modelling within a resources diagram

4.7 Data Model

According to AMBLER, a **data model** is an abstract model that describes how data are represented (structure) and accessed. Data models define data elements and their relationships for a domain of interest [Amb09]. In the following chapter, the data model of the specification technique is presented. It presents how the data is represented and accessed within the different diagrams and constructs.

The data model is depicted with the aid of UML class diagrams. The used symbols are listed in detail in Figure 4-35. This notation is used as a reference for the following figures of the data model.

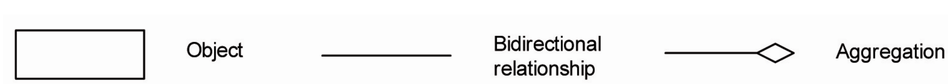

Figure 4-35: Notation of the data model

The constructs and relationship (see Chapter 4.6) introduced for the graphical description of the *partial model process sequence* are related to each other. Figure 4-36 presents an UML diagram where the associations between objects are depicted. *Process sequence diagrams* (see Figure 4-36) allow the graphical representation of the *partial model process sequence*. A process sequence consists of *processes* linked with *elements* through *sequence relationships*. Each *process* has two *sequence relationships*, but each *sequence relationship* has only one *process*. This is because the *sequence relationships* are designed to link one *process* to one or more *elements*. *Elements* can have one or two *sequence relationships*, this is because a *process sequence* starts with an *element* and finishes with an *element*. The initial and final *elements* need just one *sequence relationship*. The *process* construct is analysed with more detail at Figure 4-37 and the *element* construct at Figure 4-38. The *sequence relationship* is no further analysed because it contains no extra information rather than the succession.

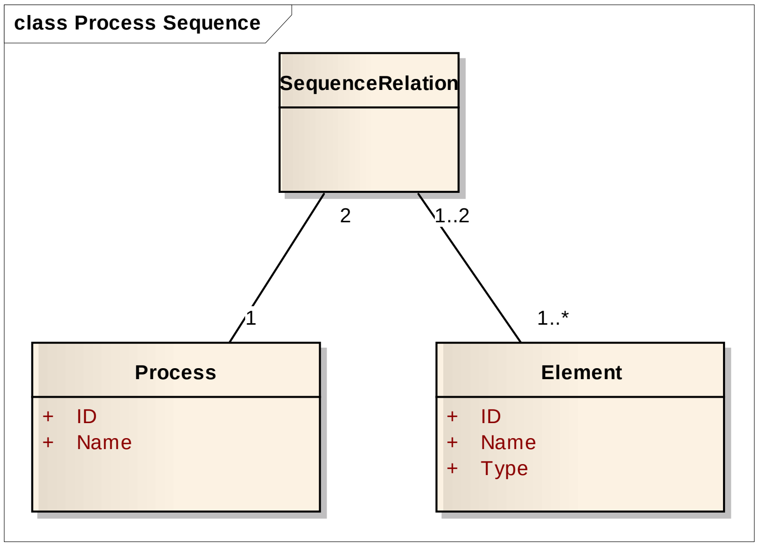

Figure 4-36: Data model of a process sequence

A *process* (in Figure 4-37) can have from zero (no subprocesses) to a finite number of subprocesses (decomposition). This is shown by the aggregation "consists of". It shows that a process construct can consist of process constructs of a lower abstraction level. This is the basis for the decomposition.

Each process has a unique ID number and a name. The **ID number** is required because each construct has a unique number that identifies it. The **name** is used to describe the type of operation that the process executes. It could be a manufacturing function, a process or a technology. Each process can have *parameters* (from zero to a finite number) and *characteristics* (from zero to a finite number). **Parameters** include technical parameters of the process, while **characteristics** (or attributes) refer to relevant information about the process, like the type of process (e.g. logistic process). For more information please refer to Chapter 4.3.2. Each process fulfils *requirements* (from zero to a finite number) and is executed by one *resource*. Nevertheless, each *resource* can execute more than one *process*.

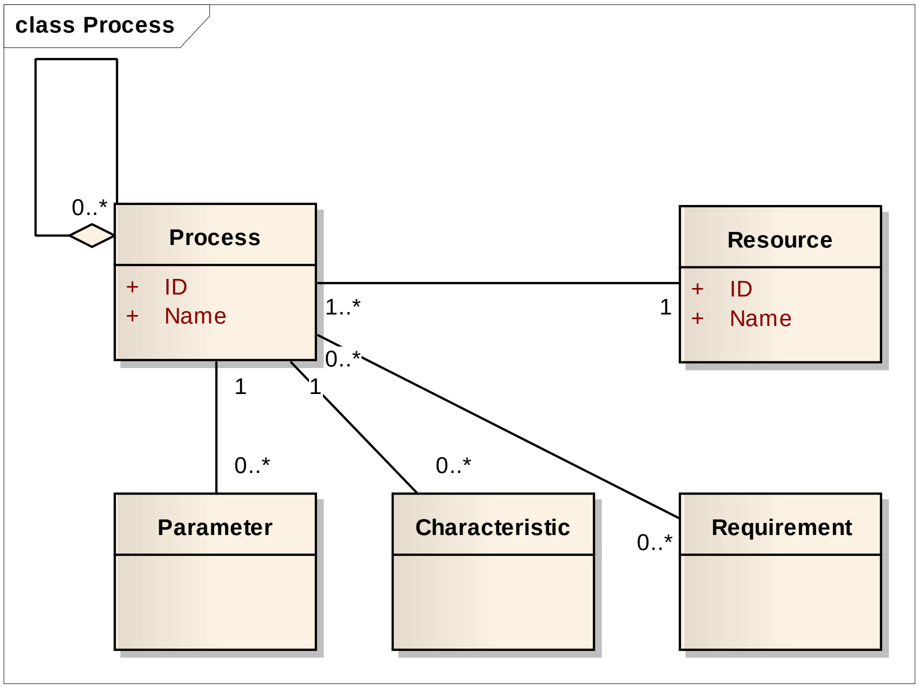

Figure 4-37: Data model of a process

The structure of an *element* is shown in Figure 4-38. Each element has an ID number, and a name. The name of the element usually is the combination of a substantive and an adjective which describes the transformation, operation or transition that the object experiences through the manufacturing trek. There are two types of elements, the *software component* and the *material element*. Each *software component* fulfils *requirements* (from zero to a finite number). Each *material element* fulfils *requirements* (from zero to a finite number) and has *shape* representations (from zero to a finite number). For more information please refer to Chapter 4.3.1.

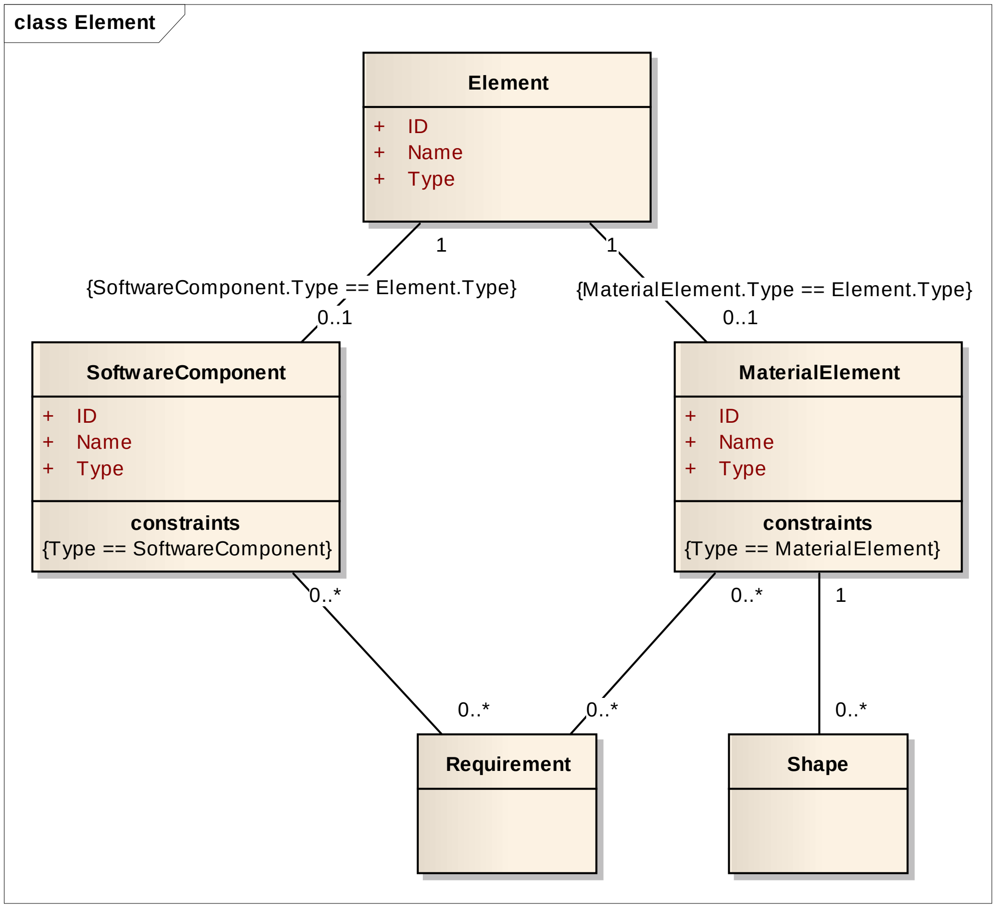

Figure 4-38: Data model of an element

The *partial model resources* needs a diagram to be depicted. Firstly, the constructs and relationships (see Chapter 4.3) introduced for the graphical description of the *partial model resources* are related to each other. Figure 4-39 presents an UML diagram where the associations between objects are depicted. *Resource diagrams* allow the graphical representation of the *partial model resources*. The simplest diagram consists of a single *resource.* This is because the simplest process has just one process and each process is made at least by one resource. Nevertheless, a resource can execute more than one process. If the process sequence has more than one process, the resources diagram can have more than one resource. If this is the case, between two resources must be a *material flow relationship* (see Chapter 4.4.3). If two or more resources work together to execute an operation (e.g. CNC machine with a drill), then an *aggregation relationship* is required (see Chapter 4.3.3).

Each resource has a unique ID number and a name. The *ID number* is required because each construct has a unique number that identifies it. The *name* is used to describe the type of resource that executes the process. It could be a worker, a machine, a tool, etc. The name of the resource usually is the name of the device. In case of the workers, an abstract denomination may be applied, e.g. Worker 23. The resource construct is analysed with more detail at Figure 4-40.

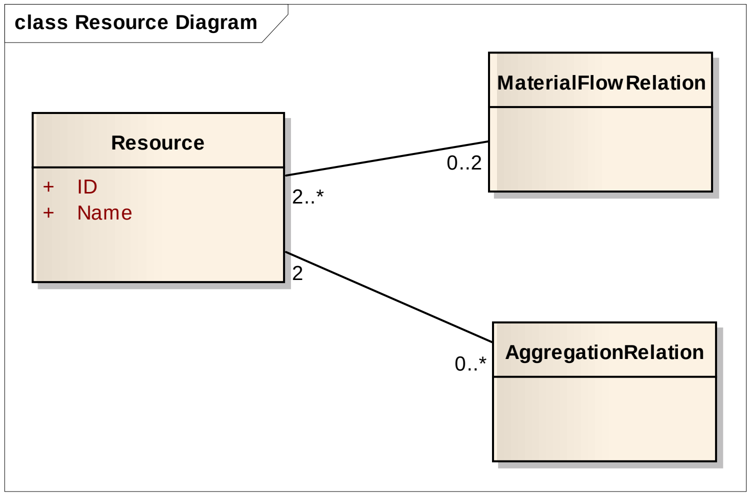

Figure 4-39: Data model of a resources diagram

In the Figure 4-40 is shown by the aggregation "consists of" that a resource construct in turn can consist of resources constructs. This property is used to increase the diagram's degree of detail in the specification technique. The aggregation "consist of" is represented with the aggregation relationship.

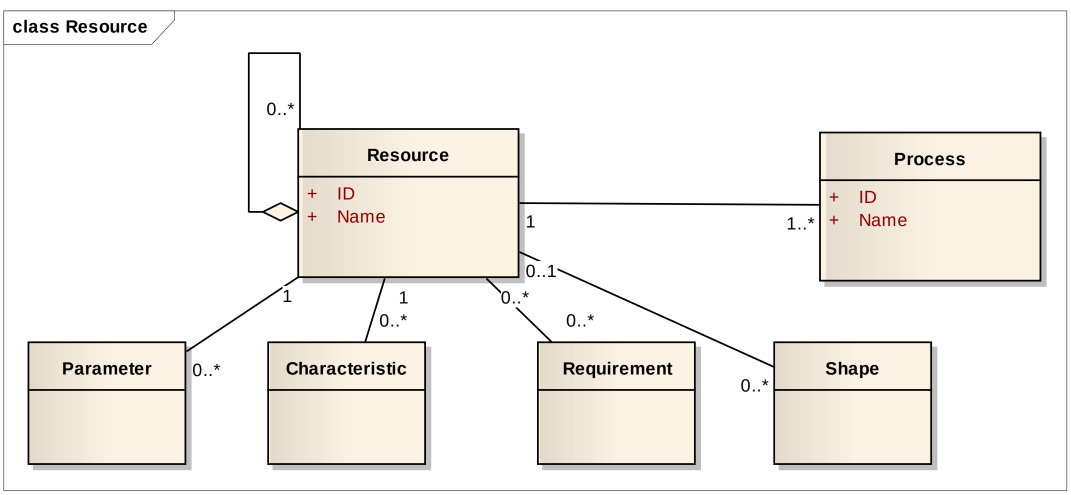

Figure 4-40: Data model of a resource construct

Each resource can have *parameters* (from zero to a finite number) and *characteristics* (from zero to a finite number). *Parameters* include technical parameters of the resources, while *characteristics* refer to relevant information about the resources. For more information please refer to Chapter 4.3.2. Each resource has *shape* representations (from zero to a finite number) and fulfils *requirements* (from zero to a finite number). For more information please refer to Chapter 4.4.1 and 4.4.4. Finally, each *resource* can execute more than one *process*.

4.8 Evaluation of the specification against the requirements

The here presented specification technique for the conceptual design of manufacturing systems successfully fulfil the requirements outlined in Chapter 2.4. The following paragraphs describe how the presented specification fulfils each of the requirements.

Requirement R1 - Holistic description of the manufacturing system's principle solution

In this work, the **basic concepts** that have to be taken into account during the conceptual design phase of manufacturing systems are identified. Besides that, a **principle approach of the conceptual design of manufacturing systems** has been developed to point out the way to specify these basic concepts within the principle solution of a manufacturing system. The approach explicitly demonstrates how these basic concepts can be specified in each partial model of the principle solution. Furthermore the fundamental **cross-references between partial models** are described. As exemplified in the application examples, the specification effectively enables an **holistic picture of the manufacturing system's principle solution** with the essential concepts included.

Requirement R2 - Description of a manufacturing system at an early stage of the product development process

The manufacturing system design is the result of the dynamic interplay between product design and manufacturing design. The here developed specification technique is based on the **principle solution of the product**. On this basis, a first concept of the manufacturing system can be depicted and, in the course of the concretization, both conceptual designs complement each other. As exemplified in the application examples, the specification technique allows a first manufacturing system's description based just on a rough idea of the product, its components and structure. This description can be further detailed during the design concretization.

Requirement R3 - Semi-formal specification technique

In this work, a consistent **semi-formal specification technique** was developed. Due to its semi-formal character, the specification allows to describe and document, effortlessly, the development process. Furthermore, it allows the general depiction of the concretization of the design. The specification counts with a **set of constructs and rules** which allow a clear depiction of a manufacturing system. To avoid contradictions or misunderstandings, a **data model** and **application examples** are provided. Furthermore, the modelling of special cases was included.

The basic concepts are specified in such a way that they can be easily interpretable even for the layman. As such, intuitive appreciation of the manufacturing concepts is possible and thus, the comprehension and linkage of various concepts among engineers of different backgrounds are feasible. In this way, the specification allows not only an integrative design but also to support constructive discussions within interdisciplinary development teams.

Requirement R4 - Domain-spanning specification technique

This specification technique **documents** a domain-spanning visualization of a manufacturing system's principle solution by means of diagrams, graphics and tables. With the division of the principle solution into **partial models**, a **balanced consideration** of the involved domains was provided.

These domains are the **requirements** to be fulfilled by the system, the **processes** to be made in order to fulfil these requirements, the **resources** to be applied within the system as well as the structure and layout of the system (**shape**). The basic concepts from these domains have been equally treated and intuitively integrated within the principle solution of a manufacturing system. As exemplified in the application examples, the basic concepts are seamlessly integrated, for instance, by means of **relationships** and **cross-references** between the partial models. As a result a comprehensible visualization of the principle solution can be engendered.

Requirement R5 - Support the iterative process of the conceptual design's development and its concretization

The partial models that conform the manufacturing system's principle solution can be **modified, concretized** or **complemented** at any moment. Furthermore, the specification's **degree of abstraction can be changed** whenever, this means that the system can be represented with different levels of detail as well as different levels of abstraction. These characteristics assist the developer during the conceptual design's typical iterative process. As exemplified in the application examples, the specification describes and documents the manufacturing system's conceptual design at any stage of its development process.

Requirement R6 - Reduce the complexity

For taming the complexity, the manufacturing system's principle solution was divided in four interrelated partial models. These partial models allow different **abstraction levels** and are **structured** according four domains: requirements, process sequence, resources and shape of a manufacturing system. The concepts of **decomposition / aggregation** and **hierarchy** were applied. These characteristics allow seeing the system as a whole, as well as in parts. The semiotic of the language was also considered. The **set of constructs has a reduced size**. The **symbols are relatively easy** to understand and are suitable as communication medium. The experience acquired in different projects where the specification was used (e.g. Transregio30[23] project) shows that the language is suitable and **comprehensible** for people of different fields.

Requirement R7 - Expandability

The specification technique was developed as a **core specification technique** with indirect expandability. This specification technique specifies a first principle solution of a manufacturing system. It includes the basic concepts and relationships for an integrative and holistic system design description. The specification allows **indirect integration.** In other words, if required, the user of this specification can extend the specification (e.g. add a view, add constructs and relationships, etc.) in order to depict specific requirements that further concretizations of the principle solution may need. These extensions may be aggregated within separated modules. Indirect integration preserves the original data model of the system and facilitates the modelling. Each new extension (view, construct, etc.) within the specification implies the enlargement of the data model and the modelling rules. With a modular expansion, misinterpretations are avoided and the model is easier to maintain and enhance.

Requirement R8 - Communication medium

The specification technique **provides manufacturing-specific concepts** and their integration **with product modelling concepts**. In this way, domain-specific concepts are provided for representing and analysing manufacturing systems. Therefore, the user does not have to reconstruct generic terms, but rather reuse the already proved domain terms. This fact increases the **comprehensibility** and **clarity of the models**. Furthermore, graphical representations allow information to be represented in a clear, understandable, visual form. They allow the visualization of complex relationships within a system. Additionally, a graphic notation leads not only to a better understanding; **it provides a common language that supports the exchange of information between the interdisciplinary design team and coordination between the design activities across the entire design process**. This facilitates a consensus that enables and enhances implementation.

---

[23] Sonderforschungsbereich SFB/TR TRR 30 "Prozessintegrierte Herstellung funktional gradierter Strukturen auf der Grundlage thermo-mechanisch gekoppelter Phänomene", in: http://www.transregio-30.com/

5 Application example

The miniature robot BeBot[24] (Figure 5-1), developed at the Heinz Nixdorf Institute, is used as a demonstrator to present and validate the developed specification technique. It serves as a test carrier for swarm intelligence, for multi-agent applications of the computer science and for the use of new manufacturing technologies. This robot is in continuous development because it constitutes the test project for modern approaches and applications, such as self-optimization, self-organization and self-coordination.

The robot is characterised by a close spatial integration of mechanics and electronics based on the technology Molded Interconnect Devices (MID). The housing is realized as a MID-component. It comprises mechanical and electrical components. The circuit traces (conductors), between the more than 100 component parts, cover the interior of the robot's chassis and create a complex, three-dimensional circuit – everything resulting in a high density of functions and a support for the miniaturization. In this way the amount of component parts is enormously small – in comparison with conventional robots.

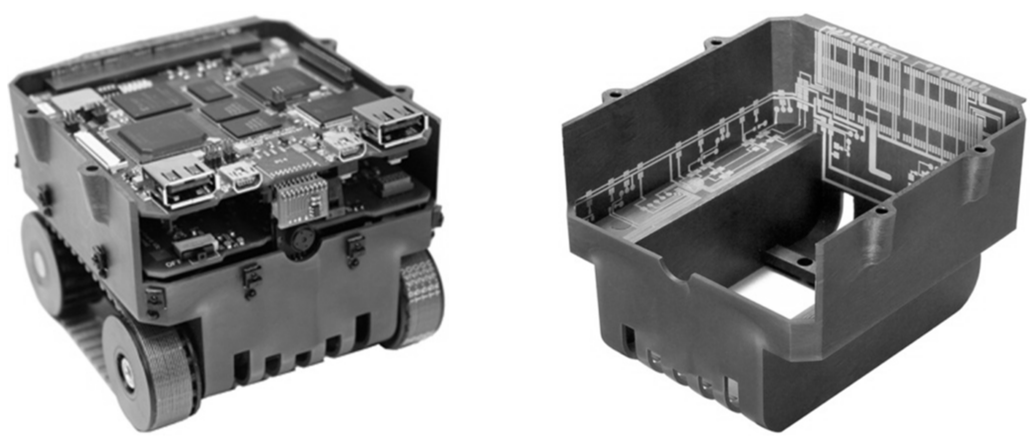

Figure 5-1: Basis module of the miniature robot BeBot (left figure) and its housing with integrated circuit traces (MID) (right figure)

The project involves experts from different disciplines, which implies different backgrounds and methodologies. It is an ambitious research project which requires an effective exchange of information and knowledge, as well as balanced consideration between the involved disciplines. Furthermore, it demands a common language that allows the conception, development, description and documentation of the project. All these requirements offer a good application field for the here developed specification technique. Consequently, the miniature robot is going to be used as demonstrator of the specification technique.

[24] http://wwwhni.uni-paderborn.de/en/priority-projects/miniaturroboter/

During the conceptual design phase of manufacturing systems, a valid first concept of the product is of paramount significance for the parallel product and manufacturing system design. This first concept of the product provides crucial information about the possible manufacturing system's tasks to be fulfilled. Following the principle approach presented in the preceding chapter, this chapter exemplifies how to specify the principle solution of the manufacturing system of the miniature robot at an early stage of the product development process. Additionally, the management of information is exemplified. In such a way, the feasibility of the aforementioned specification is validated.

Bebot's principle solution. A detail of the miniature robot's principle solution is depicted in Figure 5-2. The conceptual design of the manufacturing system is based on three partial models of the product: *requirements, active structure* and *shape*. The requirements on the miniature robot and the manufacturing system are collected within the *requirement list.* Among these requirements, it is possible to find the requirements that are made on the drive wheel (1). The drive wheel and its relation with other system elements is described within the *active structure.* The geometry and spatial relations of the drive wheel are specified within the partial model *shape* (2). Based on the *active structure* and the *shape*, the components to produce and their structural connections are derived. This information is depicted on the *building structure* (3). In this case, the *building structure* shows that the wheel and the bearing form the subassembly drive wheel. Therefore, manufacturing and assembly processes are required.

The *process sequence diagram* is developed on the basis of the components, subassemblies and assemblies of the *building structure* (4). The hierarchical structure of the building structure is also reflected in the process sequence. In order to produce the wheel, suitable processes and their sequence are selected. In this case, a *boring operation* and a *final machining* are needed to manufacture a full-fledged wheel from a blank. Finally, resources are allocated to the several processes (5). The linkage is represented as reference to the resource. Unique numbers are used to allocate processes and resources. In this case, the processes boring of the hub and final machining are allocated to the *CNC lathe* (A) and *CNC machine* (B).

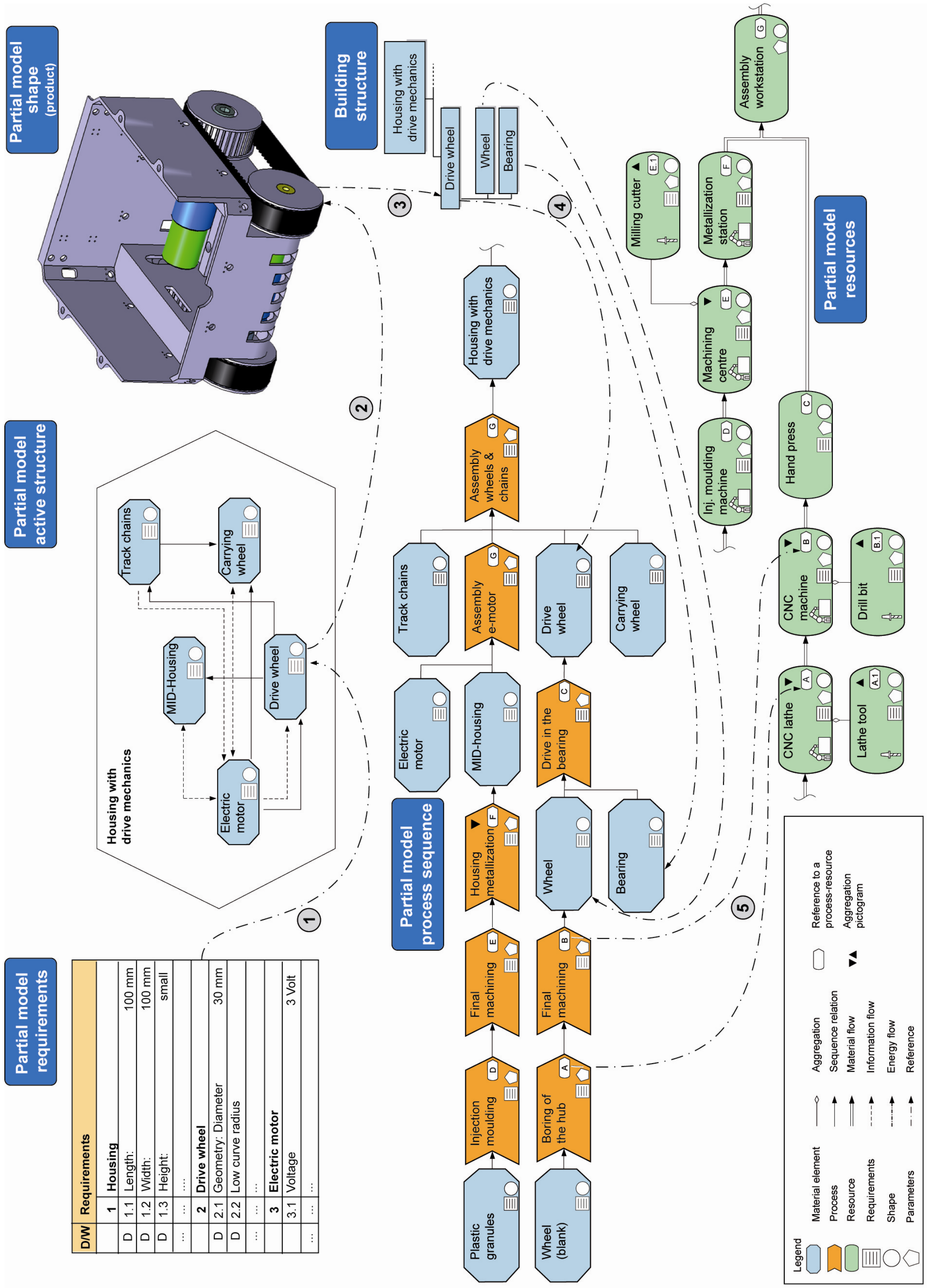

Figure 5-2: Depiction of the integrative conceptual design of the miniature robot BeBot's manufacturing system (detail)

Process sequence decomposition. Figure 5-3 presents a detail of a process decomposition. The *Process sequence diagram I* reproduces a process step within the *process sequence diagram* depicted on the partial model *process sequence* in Figure 5-2. This process step, *Housing metallization* (F), is decomposed in two process steps: *surface patterning activation* (F.1) and *put up conducting path* (F.2). These processes are depicted within the *Process sequence diagram II.*

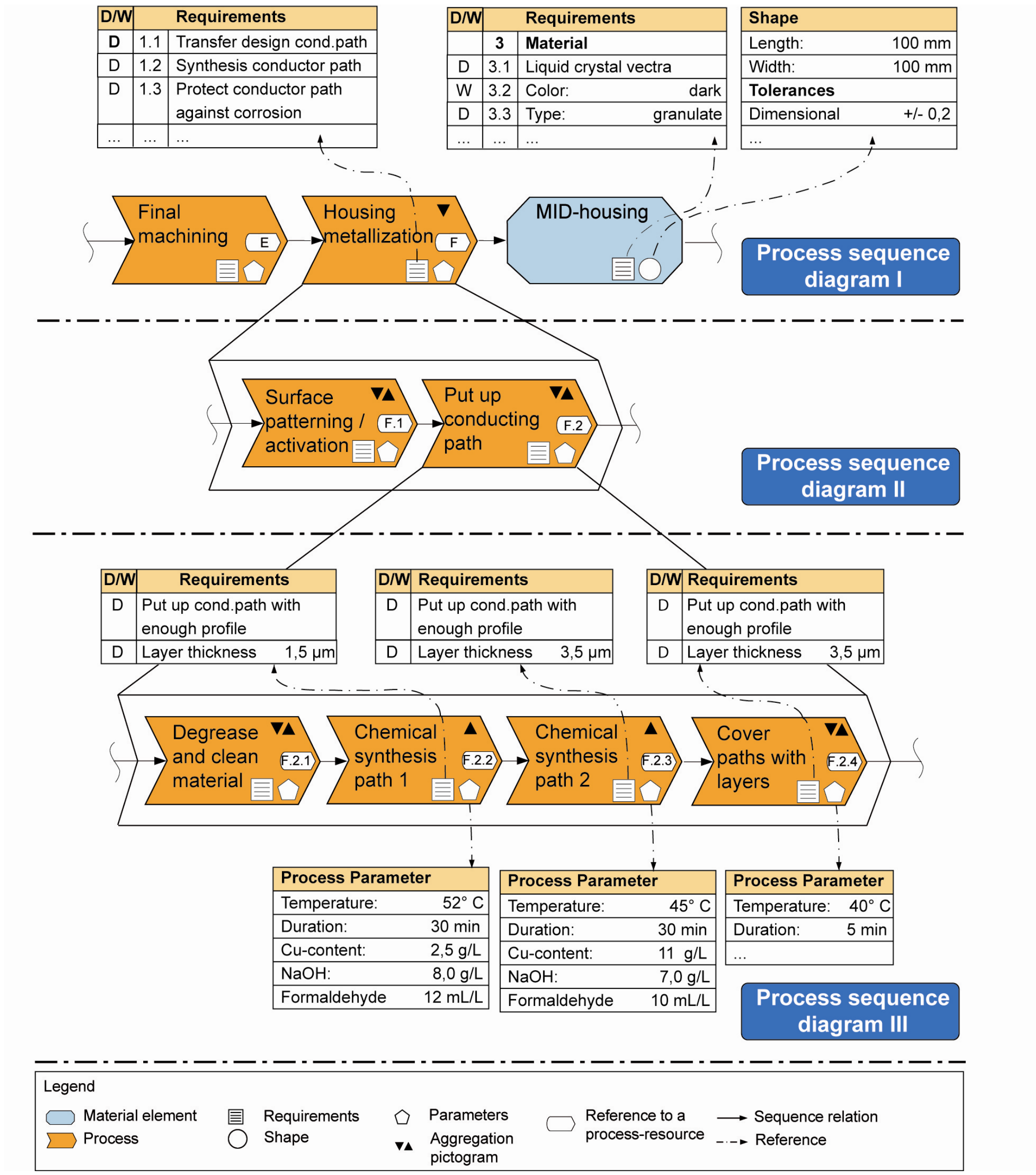

Figure 5-3: Depiction of a process sequence decomposition. Detail of the miniature robot BeBot's manufacturing system

The *put up conducting path* process is further decomposed into four process steps: *degrease and clean material* (F.2.1), *chemical synthesis path 1* (F.2.2), *chemical synthesis* *path 2* (F.2.3) and *cover paths with layers* (F.2.4). Each process is linked to the requirements list. In this way, it is easy for the user to identify which requirements are fulfilled in each process. For example, the process *chemical synthesis path 1* (F.2.2) is linked with the requirement *layer thickness 1,5 µm*, while the process step *chemical synthesis path 2* (F.2.3) is linked with the requirement *layer thickness 3,5 µm*. Additionally, each process is linked to a resource by means of the *process-resource pictogram* and to parameters and characteristics (list) by means of the *parameters pictogram*. For example the process step *cover paths with layers* (F.2.4) requires a *temperature of 40° C* with *duration of 5 minutes*.

Resources diagram. Figure 5-4 presents a detailed diagram of the resources sequence presented in Figure 5-2.

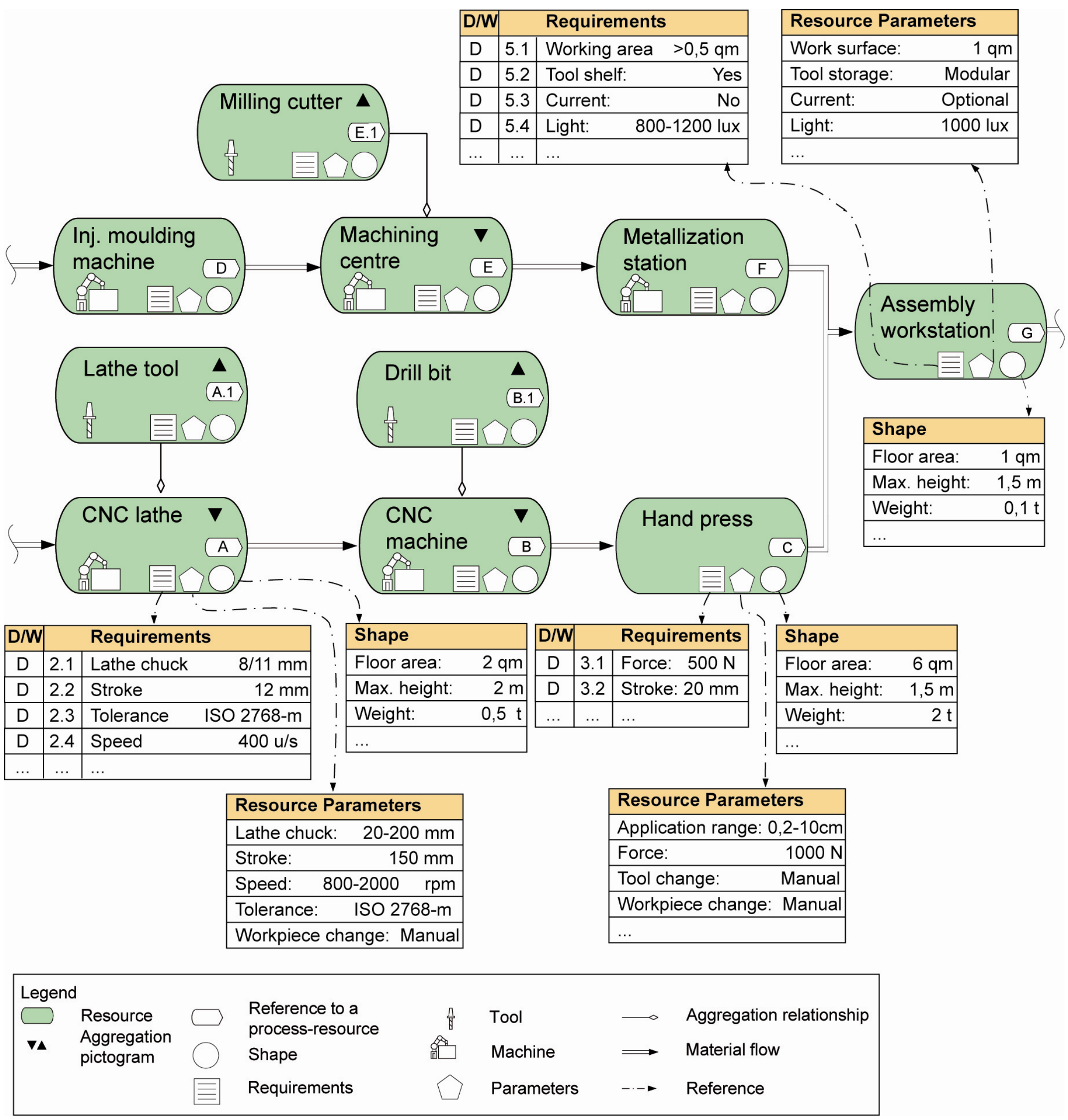

Figure 5-4: Depiction of a resources diagram. Detail of the miniature robot BeBot's manufacturing system

Each resource executes one or more processes. For this reason each resource is linked with its corresponding process or processes by means of the *process-resource pictogram.* Some resources have aggregated resources, for example the *CNC lathe* (A) uses a *Lathe tool* (A.1). This relationship is depicted by means of an *aggregation relationship* and the *aggregation pictogram.*

The resources are further concretized by means of references. For example, the resource *CNC lathe* (A) has a reference to the *requirements list*, this reference is marked with the *requirements pictogram*. Some of the requirements on the machine are its *lathe chuck* and its *speed*. Detailed information about the resource is founded within the *parameters pictogram* and the *shape pictogram*. For example, some relevant parameters and information of the resource *CNC lathe* (A) are its *tolerance* ISO 2768-m and that the *workpiece change* is manual. About its *shape*, this resource requires a *floor area* of 2 qm. It has a *height* of 2 m. and *weight* of 0,5 t.

Another example is the resource *Assembly workstation* (G). Among its parameters/characteristics we can find its *tool storage,* in this case modular and the type of *light.* In this case, the workstation requires at least 1000 lux. The reference to its shape presents a list of shape attributes that the resource has, for example this resource requires a *floor area* of 1 qm. It has a *height* of 1,5 m. and *weight* of 0,1 t.

6 Summary and Outlook

Summary

The product is strongly influenced by the production technologies; therefore, the manufacturing system design and the product must be concurrently designed. The principle solution of the product offers a rough concept of the product; it is the result of the conceptual design phase and specifies its basic structure and function. To accomplish the integration of the product and the manufacturing system, early coordination between the principle solution of the product and the manufacturing system is required.

The state of the art's results endorse the necessity of a language for the conceptual design, development, description and documentation of the principle solution of manufacturing systems at an early stage of the product development process. The manufacturing system's principle solution presents a collection of interrelated resources that perform a set of activities (processes), according to a set of requirements and needs, converting goods (inputs) into a variety of products, services and information (outputs) in the most possible cost effective way. Therefore, the required language needs to cover and represent the necessities of four domains. These domains are the requirements to be fulfilled by the system, the processes to be made in order to fulfil these requirements, the resources to be applied within the system as well as the structure and layout of the system (shape). Additionally, the interconnections between these domains must be also considered. Nevertheless, at an early stage of the product development process scarcely concrete information is available. What is more, manufacturing systems are complex systems consisting of a large number of interacting components, characterised by frequent changes in design criteria and by huge pressure to compress project delivery times. Communication and negotiation are the base of the development process. Furthermore, interdisciplinary development teams are involved and practitioners must search for innovative ways to structure the design process.

For the above mentioned reasons, a semi-formal specification technique was developed to depict a static model of a manufacturing system's principle solution. This manufacturing system's model has a top-down approach (from abstract-to-concrete) and is divided into aspects, for taming its complexity. These aspects are: requirements, process sequence, resources and shape of the manufacturing system. The aspects of the manufacturing system's principle solution are mapped on computer by partial models. These partial models were integrated to enable an holistic and integrative design. Therefore the principle solution consists of a coherent system of partial models. Through this model construction, isolated abstractions of the systems are created, enabling a comprehensive description of the system.

The partial models and their interaction represent and document the product's manufacture in the course of the system's concretization. Each partial model represents *a closed* *structured group of individual objects* which interact with other objects of other partial models. A model of a system is thus a network of objects, which are defined in terms of the relationships between each other, their internal characteristics and its behaviour.

The **PM requirements** considers the computer-internal representation of manufacturing systems' requirements. The **PM process sequence** considers the computer-internal representation of the manufacturing sequence. The **PM resources** considers the computer-internal representation of the processes' resources, e.g. machines and workers as well as the material flow that passes through them. The **PM shape of the manufacturing system** considers the computer-internal representation of the resources' shape.

Summarizing, the here presented specification technique for the conceptual design of manufacturing systems allows an holistic description of the manufacturing system's principle solution at an early stage of the product development process. It documents a domain-spanning visualization of a manufacturing system's principle solution by means of diagrams, graphics and tables. With the division of the principle solution into partial models, a balanced consideration of the involved fields was provided. Additionally, these partial models can be modified, concretized or complemented at any moment. Furthermore, the specification's degree of abstraction can be changed whenever, this means that the system can be represented with different levels of detail as well as different levels of abstraction. These characteristics assist the developer during the conceptual design's typical iterative process. In addition, the user of this specification can extend the specification in order to depict specific requirements that further concretizations of the principle solution may need. These extensions may be aggregated within separated modules. Finally, the here developed specification technique provides a common language for the manufacturing system's conceptual design stage. It supports the exchange of information between the interdisciplinary design team and coordination between the design activities across the entire design process.

Outlook

Future work includes a design methodology and/or a procedure model for the conceptual design of a manufacturing system at an early stage of the product development process. In this way, the dependencies between product and its manufacturing system can be adequately considered. Close meshing of product and manufacturing system design must be ensured in particular in the early phases of the conceptual design stage. During the conceptual design of the manufacturing system it must be checked whether the necessary production technologies and resources are available and the product can be cost-effective produced. For these reasons, additionally methods are required:

- A method for the process selection, as well as its sequence, based on the product's principle solution
- Methods for the selection of efficient production technologies and resources at an early stage of the product development process

- Evaluation methods for process sequences, technology chains and resources sequences at an early stage of the product development process
- Methods for managing the transition from the principle solution of manufacturing systems towards concretization in the domains of process planning, place of work planning, production logistics and working appliance planning

7 Bibliography

[ABB10-ol] ABB PRODUCT GUIDE: IRB 6640 & IRB 6640ID. 9. August 2010, in: http://www.abb.com/product/seitp327/df458c1c0223a1aec1257363006dd9b3.aspx

[ADG+08] ADELT, P.; DONOTH, J.; GAUSEMEIER, J.; GEISLER, J.; HENKLER, S.; KAHL, S.; KLÖPPER, B.; KRUPP, A.; MÜNCH, E.; OBERTHÜR, S.; PAIZ, C.; PORRMANN, M.; RADKOWSKI, R.; ROMAUS, C.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITKOWSKI, U.; WITTING, K.; ZNAMENSHCHYKOV, O.: Selbstoptimierende Systeme des Maschinenbaus - Definitionen, Anwendungen, Konzepte. HNI-Verlagsschriftenreihe, Band 234, Paderborn, 2008, S.20

[Agr03] AGRE, PHILIP E.: Hierarchy and History in Simon's "Architecture of Complexity". Journal of the Learning Sciences, 12: 3, 2003, Pg. 413 — 426

[AIS+77] ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M.; JACOBSON, M.; FIKSDAHL-KING, I.; Angel, A.: A Pattern Language. Oxford University Press, New York, 1977

[Amb09] AMBLER, S.W.: Data Modeling 101. Article taken from: http://www.agiledata.org/essays/dataModeling101.html, 2 July 2010

[Bec05] BECKER, T.: Prozesse in Produktion und Supply Chain optimieren. Springer-Verlag, Berlin, 2005

[Ber06] BERGER, T.: Methode zur Entwicklung und Bewertung innovativer Technologiestrategien. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftreihe, Band 176, Paderborn, 2006.

[Bun06] BUNGE, M.: Chasing Reality: Strife Over Realism. Toronto, University of Toronto Press, 2006, DOI: 10.1177/0048393108323883

[BKW+05] BRECHER, C.; KLOCKE, F.; WECK, M.; MEIDLINGER, R.; WEGNER, H.: „Bewertung von Fertigungsfolgen", Integrierte Produkt- und Prozessgestaltung – Ergebnisse des Sonderforschungsbereiches (SFB) 361 der Deutschen Forschungsgemeinschaft (DFG) an der RWTH Aachen, Eversheim, W. (ed.), Schuh, G. (ed.), Springer, Berlin, 2005.

[CFF+02] CAESAR, C.; FLEGEL, H.; FRIEDRICH, D.; GRAMS, J.; HAHN, P. O.; KLOCKE,F.; LUNG, D.; NÖKEN, S.; PÄHLER, D.; SCHMIDT, W.; TROMMER, G.; WEINTER, K.;WILLMS, H.; ZELLER, R.: Gestalten und Optimieren von Prozessketten – Umsetzung von Fertigungsalternativen heute und morgen. In: Wettbewerbsfaktor Produktionstechnik – Aachener Perspektiven. Aachener Werkzeugmaschinenkolloquium 2002. Shaker Verlag, Aachen, 2002

[Che09] LOW, C. Y.: A Methodology to Manage the Transition from the Principle Solution towards the Controller Design of Advanced Mechatronic Systems. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftreihe, Band XXX, Paderborn, 2009.

[CHN02] CISEK, R.; HABICHT, C.; NEISE, P.: Gestaltung wandlungsfähiger Produktionssysteme. ZWF 97 (2002) 9, S. 441-445.

[CVD97] CHEN D.; VALLESPIR, B.; DOUMEINGTS, G.: GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology. In: Computers in Industry Volume 33, Issues 2-3, September 1997, Pages 387-394; doi:10.1016/S0166-3615(97) 000 43-2

[Dan01] DANGELMAIER, W.: Fertigungsplanung, Planung von Aufbau und Ablauf der Fertigung. 2. Auflage, Berlin, Springer, 2001. - ISBN 3-540-42098-3

[DH02] DAENZER, W.F.; HUBER, F.: Systems Engineering ─ Methoden und Praxis. 11. durchgesehene Auflage, Zürich: Verlag Industrielle Organisation, 2002

[DIN69900] DIN DEUTSCHES INSTITUT FÜR NORMUNG E. V. (Hrsg.): DIN-Norm 69 900 „Netzplantechnik“. Beuth Verlag, Berlin, 1987

[DIN69900-1] DIN Deutsches Institut für Normung e.V. (Hrsg.): DIN 69900-1, Netzplantechnik - Begriffe, Teil 1. August 1987

[DIN69902] DIN DEUTSCHES INSTITUT FÜR NORMUNG E.V. (Hrsg.): DIN 69902 „Projektwirtschaft Einsatzmittel - Begriffe“. Beuth Verlag, Berlin, 1987

[DIN8580] DIN DEUTSCHES INSTITUT FÜR NORMUNG E. V. (Hrsg.): DIN-Norm 8580 „Fertigungsverfahren – Begriffe, Einteilung“. Beuth-Verlag, Berlin, 2003

[Enc10] ENCARTA WORLD ENGLISH DICTIONARY (Hrsg.). http://uk.encarta.msn.com/encnet/features/dictionary/dictionaryhome.aspx

[Eve97] EVERSHEIM,W.: Organisation in der Produktionstechnik Teil 3 – Arbeitsvorbereitung. 3. Auflage, Springer Verlag, Berlin, 1997

[Fal00] FALLBÖHMER, M.: Generieren alternativer Technologieketten in frühen Phasen der Produktentwicklung. Dissertation an der RWTH Aachen. Berichte aus der Produktionstechnik, Band 23/2000. Shaker Verlag, Aachen, 2000

[FGK+04] FRANK, U.; GIESE, H.; KLEIN, F.; OBERSCHELP, O.; SCHMIDT, A.; SCHULZ, B.; VÖCKING, H.; WITTING, K.: Selbstoptimierende Systeme des Maschinenbaus - Definitionen und Konzepte. HNI-Verlagsschriftenreihe, Band 155, Paderborn, 2004, S. 55-58

[FMP+01] FLAKE, S.; MUELLER, W.; PAPE U.; RUF, J.: Analyzing Timing Constraints in Flexible Manufacturing Systems. In: Proceedings of Intelligent Manufacturing Systems (IAM'2001). Dubai, March 2001

[FMS06] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: OMG Systems Modeling Language (OMG SysML™). Tutorial. INCOSE, 11 July 2006. Online verfügbar unter http://www.omgsysml.org/.

[Fra06] FRANK, U.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 175, Paderborn, 2006

[FR98] FRANK, U.; RITTGEN, P.: Zur Anreicherung von Modellierungsmethoden mit domänenspezifischem Wissen: Chancen und Herausforderungen der Unternehmensmodellierung. In: POHL, K.; SCHÜRR, A.; VOSSEN, G. (Hrsg.): Modellierung, Bd. 9 von CEUR Workshop Proceedings, CEUR-WS.org 1998.

[Fra99] FRANK, U.: Visual Languages for Enterprise Modelling. Arbeitsbericht, Arbeitsberichte des Instituts für Wirtschafts- und Verwaltungsinformatik, Universität Koblenz-Landau, Nr. 18 1999.

[Fuk07] FUKUDA, S.: Be Lazy: A Motto for New Concurrent Engineering. In: LOUREIRO, G.; CURRAN, R.: Complex Systems Concurrent Engineering- Collaboration, Technology Innovation and Sustainability. http://dx.doi.org/10.1007/978-1-84628-976-7

[FSB+08] FELDMANN, K.; SCHMUCK, T.; BROSSOG, M.; DREYER, J.: „Beschreibungsmodell zur Planung von Produktionssystemen“, wt Werkstattstechnik online, Vol. 98, Springer-VDI-Verlag, 2008, pp. 156-162.

[GA09] GROTE, K.-H.; ANTONSSON, E. K.: Springer handbook of mechanical engineering. Springer (Springer handbooks), Berlin, 2009

[GBR10] GAUSEMEIER, J.; BRANDIS, R.; REYES-PEREZ, M.: A Specification Technique For The Integrative Conceptual Design Of Mechatronic Products And Production Systems. In: Proceedings of the International Design Conference - DESIGN 2010, Mai 2010

[GEK01] GAUSEMEIER, J.; EBBESMEYER, P.; KALLMEYER, F.: Produktinnovation – Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag, München, 2001

[GF99] GAUSEMEIER, J.; FINK, A: Führung im Wandel – Ein ganzheitliches Modell zur zukunftsorientierten Unternehmensgestaltung. Hanser, München, 1999

[GFD+09] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL.S.: Specification technique for the description of self-optimising mechatronic systems. In: Research in Engineering Design, Vol. 20, No. 4, 2009, pp 201-223

[GFD+08] GAUSEMEIER, J; FRANK, U; DONOTH, J; KAHL, S: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus - Teil 1. Konstruktion, Juli/August, Heft 7/8, Springer-VDI-Verlag, Düsseldorf, 2008, S. 59-66

[GGS+07] GAUSEMEIER J.; GIESE, H.; SCHÄFER, W; AXENATH, B.; FRANK, U.; HENKLER, S; POOK, S.; TICHY, M.: Towards the Design of Self-Optimising Mechatronic Systems: Consistency between Domain-Spanning and Domain-Specific Models. Proc. of the 16th International Conference on Engineering De-sign (ICED), Paris, France, 2007

[GK06] GIENKE, H.; KÄMPF, R.; Praxishandbuch Produktion– Innovatives Produktionsmanagement: Organisation,Konzepte, Controlling. Carl Hanser Verlag, München, 2006

[GPW09] GAUSEMEIER, J.; PLASS, C.; WENZELMANN, C.: Zukunftsorientierte Unternehmensgestaltung – Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen, Carl Hanser Verlag, München, 2009.

[GSG+09] GAUSEMEIER, J.; SCHÄFER, W.; GREENYER, J.; KAHL, S.; POOK, S.; RIEKE, J.: Management of Cross-Domain Model Consistency During the Development of Advanced Mechatronic Systems. In: Proceedings of the 17th International Conference on Engineering Design (ICED'09) (Margareta~Norell Bergendahl, Martin Grimheden, and Larry Leifer, eds.), vol. 6, (University of Stanford, CA, USA), pp. 1-12, Design Society, August, 2009

[HHK+08] HENKLER, S.; HIRSCH, M.; KAHL, S.; SCHMIDT, A.: Continuous development of a self-optimising airgap adjustment system for autonomous vehicles. In: Proceedings of IDETC/CIE 2008 ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Band ., 3. - 8. United Kingdom, August, 2008

[Hau06] HAUSE, M.: The SysML Modelling Language. 5th European Systems Engineering Conference, September 18-20, Edinburgh, Scotland, 2006

[HJ02] HOUSHMAND, M.; JAMSHIDNEZHAD, B.: Conceptual design of lean production systems through an axiomatic design. Proceedings of ICAD2002, second international conference on axiomatic design, Cambridge, MA, 2002

[HH91] HELD, G.; HAGGENMÜLLER, R.: GRAPES language description. Syntax, semantics and grammar of GRAPES-86. Berlin: Siemens Aktienges., 1991

[HRM07] HUANG, E.; RAMAMURTHY, R.; MCGINNIS, L.F.: System And Simulation Modelling Using SysML. Proceedings of the 2007 Winter Simulation Conference S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds., 2007, Pg. 796-803

[IDE10-ol] IDEF INTEGRATED DEFINITION METHODS: IDEF Familiy of methods. 1. February 2010, in: http://www.idef.com/

[IDE10a-ol] IDEF INTEGRATED DEFINITION METHODS: IDEF0 Function modelling method. 1. February 2010, in: http://www.idef.com/IDEF0.htm

[IDE10b-ol] FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION 183 (FIPSPUB 183): Integration Definition For Function Modelling (IDEF0). Issued by the National Institute of Standards and Technology, 1993. PDF version can be downloaded from: http://www.idef.com/pdf/idef0.pdf, 1 February 2010

[IDE10c-ol] IDEF INTEGRATED DEFINITION METHODS: IDEF3 Process description capture method. 1. February 2010, in: http://www.idef.com/IDEF3.htm

[IDE10d-ol] MAYER, R. J., MENZEL, C. P.; PAINTER, M. K.; DEWITTE, P. S.; BLINN T.; PERAKATH B.: Information Integration For Concurrent Engineering (Iice) Idef3 Process Description Capture Method Report. Knowledge Based Systems, Inc., Texas, 1995. PDF version can be downloaded from: http://www.idef.com/pdf/Idef3_fn.pdf, 1 February 2010

[INC10-ol] INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE) (Hrsg.): A Consensus of the INCOSE Fellows. 9. August 2010, in: http://www.incose.org/practice/fellowsconsensus.aspx

[INC10a-ol] INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE) (Hrsg.): A What is Systems Engineering? 9. August 2010, in: http://www.incose.org/practice/whatissystemseng.aspx

[JR00] JOSLYN, C.; ROCHA, L.: Towards semiotic agent-based models of socio-technical organisations, Proc. AI, Simulation and Planning in High Autonomy Systems (AIS 2000) Conference, Tucson, Arizona, 2000, pp. 70-79.

[KM07] KWON K.; MCGINNIS, L.F.; SysML-based Simulation Framework for Semiconductor Manufacturing. Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering Scottsdale, AZ, USA, Sept 22-25, 2007, Pg. 1075-1080.

[Kan92] KANAWATY, G.: Introduction to work study. 4th (rev.) ed. Geneva: International Labour Office, 1992

[Lan00] LANGLOTZ, G.: Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte. Dissertation, Institut für Rechneranwendung in Planung und Konstruktion, Universität Karlsruhe, Shaker-Verlag, Band 2/2000, Aachen

[LAN09] LANKHORST, M.: Enterprise Architecture at Work- Modelling, Communication and Analysis. 2nd ed., 2009,Springer Verlag 352 pp. ISBN: 978-3-642-01309-6

[Lin01] LINCK, J.: A Decomposition-Based Approach for Manufacturing System Design, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA., 2001

[LS06] LEE, Q.; Snyder, B.: The Strategos guide to Value Stream Mapping and Process Mapping-Genesis of Manufacturing strategy. E-Book, published by Strategos Inc., Kansas City, 2006

[Mer10-ol] Merriam Webster Dictionary. 9 August 2010 in: http://www.merriam-webster.com/ dictionary

[Mic06] MICHELS, J.M.: Integrative Spezifikation von Produkt- und Produktionssystemkonzeptionen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 196, Paderborn, 2006

[NAB04] NÄSER, P.; ACKERMANN, J.; BAUM, H.:Das Sichtenkonzept als Modellierungsparadigma. In: Industrie Management. 20 (2004), Gito mbH Verlag, Berlin, 2004. Pg.36-39.

[NHR+08] NYHUIS, P.; HEINEN, T.; REINHART, G.; RIMPAU, C. et al.: Wandlungsfähige Produktionssysteme – Theoretischer Hintergrund zur Wandlungsfähigkeit von Produktionssystemen. wt Werkstattstechnik online 98 (2008) Nr. 1/2, S. 85–91. Internet: www.werkstattstechnik.de. Düsseldorf: Springer-VDI-Verlag

[NIS04-ol] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (Hrsg.): Directed acyclic graph. 9.August 2010, in: http://www.itl.nist.gov/div897/sqg/dads/HTML/directAcycGraph.html

[Oxf10-ol] OXFORD DICTIONARIES (Hrsg.). 9. August 2010 in: http://oxforddictionaries.com/

[OMG07] OBJECT MANAGEMENT GROUP: OMG Systems Modeling Language (OMG SysML™) V1.0 ─ OMG Available Specification, 2007, in: http://www.omgsysml.org/

[OMG09-ol] OBJECT MANAGEMENT GROUP: OMG Systems Modeling Language The Official OMG SysML site, February 2009, in: http://www.omgsysml.org/

[OMG09a-ol] OBJECT MANAGEMENT GROUP: OMG Systems Modeling Language. Introduction to OMG's Unified Modeling Language (UML). February 2010, in: http://www.omg.org/gettingstarted /what_is_uml.htm

[PB97] PAHL, G.; BEITZ, W.: Konstruktionslehre - Methoden und Anwendung. Bd. 4. 1. Springer-Verlag, Berlin, 1997

[PBF+07] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Engineering Design – A Systematic Approach. 3rd English edition, Springer Verlag, London, 2007.

[PL08] PONN, J.; LINDEMANN, U.: Konzeptentwicklung und Gestaltung technischer Produkte. Springer-Verlag, Berlin, 2008

[Rop99-ol] ROPOHL, G.: Philosophy of socio-technical systems, in: Society for Philosophy and Technology, Spring 1999, Volume 4, Number 3, 1999. 9, August, 2010, in: http://scholar.lib.vt.edu/ejournals/SPT/v4_n3html/ROPOHL.html

[Rot00] ROTH, K.-H.: Konstruieren mit Konstruktionskatalogen. Band 1, Konstruktionslehre, Springer Verlag, Berlin, 2000.

[REF87] REFA - Verband für Arbeitsstudien und Betriebsorganisation e.V. (Hrsg.): Methodenlehre der Betriebsorganisation - Planung und Gestaltung komplexer Produktionssysteme. München, Carl Hanser Verlag, 1987. -ISBN 3-446-15057-9

[REF91] REFA - VERBAND FÜR ARBEITSSTUDIEN UND BETRIEBSORGANISATION E.V (Hrsg.): Methodenlehre der Betriebsorganisation - Planung und Steuerung Teil 2. Programm und Auftrag, Materialplanung und -steuerung, Kapazitätsplanung und -steuerung, Personalplanung und -steuerung, Betriebsmittelplanung und -steuerung. München, Carl Hanser Verlag, 1991

[REF91a] REFA – VERBAND FÜR ARBEITSSTUDIEN UND BETRIEBSORGANISATION E.V. (Hrsg.): Methodenlehre der Betriebsorganisation – Planung und Steuerung Teil 3. Durchlaufzeit- und Terminermittlung, Arbeitsunterlagen, Werkstattsteuerung. Carl Hanser Verlag, München, 1991

[RSW99] ROTHER, M.; SHOOK, J.; WOMACK, J.: Learning to See: Value-Stream Mapping to Create Value and Eliminate Muda: Value Stream Mapping to Add Value and Eliminate Muda. Lean Enterprise Institute, USA, 1. February 1999.

[Sch00] SCHMIDTMANN, A.: Eine Spezifikationssprache für die Fertigungslenkung. Dissertation, Heinz Nixdorf Institut, Universität-GH Paderborn, HNI-Verlagsschriftreihe, Paderborn, Band 69, 2000

[Sch01] SCHWARZE, J.: Projektmanagement mit Netzplantechnik. 8. Auflage, Herne, Verlag Neue Wirtschafts-Briefe, 2001. -ISBN 3-482-56068-4

[SCP98] SUH, N. P.; COCHRAN, D. S.; PAULO, C. L.: Manufacturing system design. Annals of the CIRP, 1998, 47(2), 627–639

[SJ08] SPRENGER, J.; JÜNG, J. : Enterprise Modelling in the Context of Manufacturing - Outline of an Approach Supporting Production Planning, ICB-Research Report, Institut für Informatik und Wirtschaftsinformatik (ICB), Nr. 23, 2008

[ST97] SENECHAL, O.; TAHON, C.: A modelling approach for production costing and continuous improvement of manufacturing processes. In: Production Planning and Control 8 (1997) 8, S. 731-742

[Ste07] STEFFEN, D: Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftreihe, Band 207, 2007, Paderborn

[Suh01] SUH, N.P.: Axiomatic Design ─ Advances and Applications. Oxford University Press, New York, 2001

[SW04] SCHENK, M.; WIRTH, S.: Fabrikplanung und Fabrikbetrieb. Methoden für die wandlungsfähige und vernetzte Fabrik. Springer, Berlin, 2004

[Trä07] TRÄCHTLER, A.: Modellbasierter Entwurf mechatronischer Fahrwerkssysteme. 5. Paderborner Workshop „Entwurf mechatronischer Systeme", 22.-23. März 2007, Paderborn, HNI-Verlagsschriftenreihe, Band 210, 2007, S. 3-13

[Ver06] VERNADAT, F.: The CIMOSA Languages. In: Handbook on Architectures of Information Systems Second Edition Springer Berlin Heidelberg April, 2006, Seiten 251-272; DOI10.1007/3-540-26661-5_11

[VR08] VALKMAN, Y.; RYKHALSKY, A.Y: Architecture of Model Parametric Space: Hierarchy in Simon's Architecture of Complexity. In: International Conference on Inductive Modelling, Kyiv, September 2008, Pg. 58-59

[VDI2206] VDI-GESELLSCHAFT ENTWICKLUNG KONSTRUKTION VERTRIEB (Hrsg.): VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme", Beuth-Verlag, Berlin, 2004

[VDI2860] VDI-GESELLSCHAFT PRODUKTIONSTECHNIK (ADB) (Hrsg.): VDIRichtlinie 2860 „Montage und Handhabungstechnik – Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole". Beuth Verlag, Berlin, 1990

[VDI/VDE3682] VDI/VDE-GESELLSCHAFT MESS- UND AUTOMATISIERUNGSTECHNIK, (HRSG.): VDI/VDE 3682 „Formalisierte Prozessbeschreibungen". Beuth-Verlag, Berlin, 2005

[War93] WARNECKE, H.-J.: Der Produktionsbetrieb 1. Organisation, Produkt, Planung. 2. Auflage. Springer Verlag, Berlin, 1993

[WBB+94] WILLIAMS, T.J.; BERNUS, P.; BROSVIC, J.; CHEN, D.; DOUMEINGTS, G.; NEMES, L.; NEVINS, J.L.; VALLESPIR, B.; VLIETSTRA, J.; ZOETEKOUW, D.: Architectures for integrating manufacturing activities and enterprises. In: Computers in Industry, Volume 24, Issues 2-3, September 1994, Pages 111-139; doi:10.1016/0166-3615(94)90016-7

[Web07] WEBER, CH.: Looking at "DFX" and "Product Maturity" from the Perspective of a New Approach to Modelling Product and Product Development Processes. In: The future of product development. Proceedings of the 17th CIRP Design Conference. Springer, Berlin, 2007

[Wes06] WESTKÄMPER, E. : Einführung in die Organisation der Produktion. Berlin: Springer (Springer-Lehrbuch), 2006

[Wil08] WILLMS, H.: Methodisches System zur Auslegung von kostenoptimalen und prozessstabilen Fertigungsverkettungen. Dissertation an der RWTH Aachen, Lehrstuhl für Technologie der Fertigungsverfahren, Fakultät für Maschinenwesen,2008. URL: http://darwin.bth.rwth-aachen.de/opus3/volltexte/2008/2484/

[Wie97] WIENDAHL, H.-P.: Betriebsorganisation für Ingenieure. 4. Auflage, Carl Hanser Verlag, München, 1997

[Wu92] WU, B.: Manufacturing Systems Design and Analysis. Chapman and Hall, 1992

[ZMR05] ZAEH, M., MUELLER, N., RIMPAU, C.: A Holistic Framework for Enhancing the Changeability of Production Systems. In: ZAEH, M. (ED.), REINHART, G. (ED.): 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Prduction (CARV 2005). Herbert Utz Verlag, München, 2005

[3DC09-ol] 3D CAD BROWSER (Hrsg.): Robot Fanuc-S430i [NURBS]. 9. August 2010, in: www.3dcadbrowser.com/preview.aspx?ModelCode=29013

[3DC09a-ol] 3D CAD BROWSER (Hrsg.): ABB IRB6640 Robot [NURBS]. 9. August 2010, in: http://www.3dcadbrowser.com/preview.aspx?ModelCode=24492

Curriculum vitae

Personal data

Name:	Mariana Reyes Pérez
Date of birth:	17th. November 1981
Birthplace:	Mexico City, Mexico
Marital status:	Married
Nationality:	Mexican

Education

09/1988 – 07/1994	„La Salle“, Primary School in Mexico City
09/1994 – 07/1997	„La Salle“, Secondary School in Mexico City
09/1997 – 07/2000	„La Salle del Pedregal“, High School in Mexico City
08/2000 – 05/2005	ITESM CCM University in Mexico City
	Degree: Engineer in Mechatronics

Professional Experience

01/2003 – 01/2004	Research assistant of Dr. Pedro Ponce (SNI, National Researcher)
01/2003 – 10/2004	Fuzzy logic manual in collaboration with Dr. Pedro Ponce
01/2005 – 06/2005	Manufacturing Engineer in BOCAR Services (Mexico)
07/2005 – 01/2006	Plant Coordinator of Continuous Improvement in BOCAR Services (Mexico)
10/2007 – 09/2010	Research assistant at Heinz Nixdorf Institut, Prof. Dr.-Ing. Jürgen Gausemeier

Wald, 4th. October 2010

Das Heinz Nixdorf Institut – Interdisziplinäres Forschungszentrum für Informatik und Technik

Das Heinz Nixdorf Institut ist ein Forschungszentrum der Universität Paderborn. Es entstand 1987 aus der Initiative und mit Förderung von Heinz Nixdorf. Damit wollte er Ingenieurwissenschaften und Informatik zusammenzuführen, um wesentliche Impulse für neue Produkte und Dienstleistungen zu erzeugen. Dies schließt auch die Wechselwirkungen mit dem gesellschaftlichen Umfeld ein.

Die Forschungsarbeit orientiert sich an dem Programm „Dynamik, Mobilität, Vernetzung: Auf dem Weg zu den technischen Systemen von morgen". In der Lehre engagiert sich das Heinz Nixdorf Institut in vielen Studiengängen der Universität. Hier ist das übergeordnete Ziel, den Studierenden die Kompetenzen zu vermitteln, auf die es in der Wirtschaft morgen ankommt.

Heute wirken am Heinz Nixdorf Institut sieben Professoren mit insgesamt 200 Mitarbeiterinnen und Mitarbeitern. Etwa ein Viertel der Forschungsprojekte der Universität Paderborn entfallen auf das Heinz Nixdorf Institut und pro Jahr promovieren hier etwa 30 Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler.

Heinz Nixdorf Institute – Interdisciplinary Research Centre for Computer Science and Technology

The Heinz Nixdorf Institute is a research centre within the University of Paderborn. It was founded in 1987 initiated and supported by Heinz Nixdorf. By doing so he wanted to create a symbiosis of computer science and engineering in order to provide critical impetus for new products and services. This includes interactions with the social environment.

Our research is aligned with the program "Dynamics, Mobility, Integration: En-route to the technical systems of tomorrow." In training and education the Heinz Nixdorf Institute is involved in many programs of study at the University of Paderborn. The superior goal in education and training is to communicate competencies that are critical in tomorrows economy.

Today seven Professors and 200 researchers work at the Heinz Nixdorf Institute. The Heinz Nixdorf Institute accounts for approximately a quarter of the research projects of the University of Paderborn and per year approximately 30 young researchers receive a doctorate.