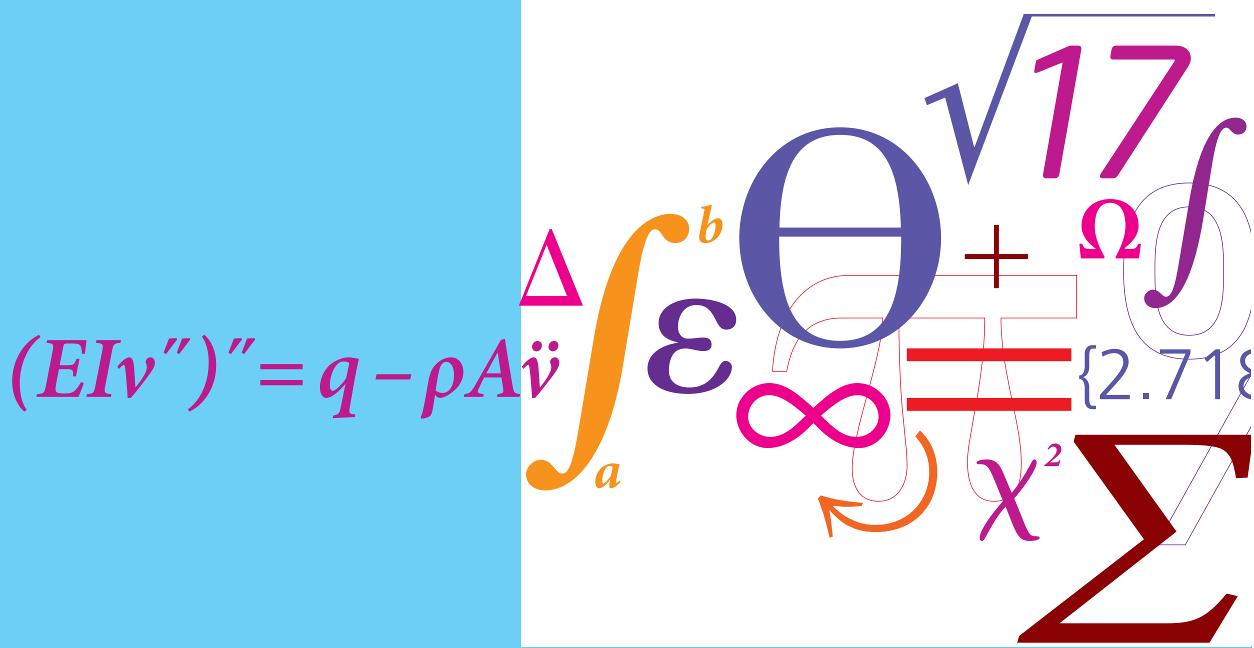
ARCHITECTURE DESCRIPTIONS

- A Contribution to Modeling of Production System Architecture

PhD Thesis



Allan Dam Jepsen

DCAMM Special Report No. S186
September 2014

DTU Mechanical Engineering
Department of Mechanical Engineering

Abstract

The subject of this PhD dissertation is architecture-centric design and the description of production system architecture.

Companies are facing demands for the development and production of new products at an ever increasing rate, as the market life of products decreases and the rate at which customers demand new product features and performance accelerates. Many of these companies are seeking to keep pace with market demands and the pressures of low cost production in other countries by adopting an architecture-centric or platform based approach to the design of their production systems. As companies seek to put the architecture at the center of design activities and let it be a focal point throughout the system life-cycle, they discover a need to change their view of the system design and how they handle it. Applying an architecture-centric approach to production system design requires a proper understanding of the architecture phenomenon and the ability to describe it in a manner that allow the architecture to be communicated to and handled by stakeholders throughout the company.

Despite the existence of several design philosophies in production system design such as Lean, that focus on the underlying principles of a production system's design; and despite the existence of established architecture and platform theories and practices within product design, there is still a need for a better understanding of the architecture phenomenon itself, and certainly how it applies within production system design. This research contributes to the vocabulary and understanding of the architecture phenomenon. A conceptual framework is provided which allows for conceptualization of the architecture phenomenon, and how it applies within production system design.

To aid the companies in the operational design and handling of production system architecture, research is conducted into the description of production system architecture, including what an architecture description contains in general and what it should describe for production systems specifically. The contribution in this area of research consists of three parts. First, a conceptual model of architecture descriptions is established based on the ISO/IEC/IEEE 42010 standard. Secondly, the stakeholders and architecture related concerns of relevance for descriptions of production system architectures are investigated, and requirements for the descriptive capabilities of production system architecture descriptions are formulated. Thirdly, a reference architecture framework is suggested. The reference architecture framework will allow system stakeholders to describe the architecture of production systems based on a common set of viewpoints. The viewpoints provide a set of model kinds to frame select architecture related concerns relating to the production capability and the design of the technical system.

With the contribution to architecture description there follows a need to support exchange and processing of architecture information within a diverse set of stakeholder domains and tools in the production system life cycle. To support such activities, a contribution is made to the identification and referencing of production system elements within architecture descriptions as part of the reference architecture framework. The contribution consists of a reference designation system based on the ISO/IEC 81346 standard series. The system allows for identification and referencing of the system elements through identifiers generated based on the compositional structures present in the production system.

Keywords: production system, system architecture, architecture description, reference designation system

Resumé (In Danish)

Temaet for denne PhD-afhandling er arkitektur-centreret design og beskrivelsen af arkitektur for produktionssystemer.

I takt med at produkters levetid på markedet falder og kunders efterspørgsel efter nye produktegenskaber og ydeevne stiger, står flere virksomheder overfor krav om hurtigere udvikling og produktion af nye produkter. Mange af disse virksomheder søger at holde trit med markedets krav og presset fra andre lande med lavere produktionsomkostninger, ved at adoptere en arkitektur-centreret eller platformsbaseret tilgang til design af deres produktionssystemer. Efterhånden som disse virksomheder fokuserer mere på arkitekturen i deres designaktiviteter og lader den være omdrejningspunktet i hele systemets livscyklus, opdager de et behov for at ændre deres syn på systemets design og hvordan det skal håndteres. Design af produktionssystemer ift. en arkitektur-centreret designfilosofi kræver en korrekt forståelse af arkitekturfænomenet samt evnen til at beskrive arkitekturen på en måde der lader arkitekturen kommunikeres til og håndteres af interessenter i hele virksomheden.

Selv om der findes flere designfilosofier inden for design af produktionssystemer, såsom Lean, der fokuserer på de grundlæggende principper for et produktionssystems design, og på trods af eksistensen af etablerede arkitektur- og platformsteorier og design metoder, er der stadig et behov for en bedre forståelse af selve arkitekturfænomenet og hvordan det kan anvendes inden for design af produktionssystemer. Dette forskningsprojekt bidrager til termer og forståelse ift. arkitekturfænomenet, gennem et konceptuelt rammeværk, som gør det muligt at uddybe arkitekturkonceptet og hvordan det kommer til udtryk inden for design af produktionssystemer.

For at hjælpe virksomheder med operationelt design og håndtering produktionssystemarkitektur, bidrager forskningen ydermere til beskrivelse af produktionssystemarkitektur. Det beskrives hvad en arkitekturbeskrivelse generelt indeholder og hvad den specifikt bør indeholde når arkitekturen for et produktionssystem skal beskrives. Bidraget hertil består af tre dele. Først og fremmest etableres der en konceptuel model for arkitekturbeskrivelser baseret på ISO/IEC/IEEE 42010 standarden. Dernæst udforskes de relevante interessenter for beskrivelser af produktionssystemers arkitektur og deres interesser ift. arkitekturen. Endeligt udvikles et rammeværk der kan bruges som grundlag for beskrivelse af produktionsarkitekturer. Rammeværket vil gøre det muligt for systemets interessenter at beskrive arkitekturer for produktionssystemer baseret på et fælles sæt af såkaldte viewpoints. De to inkluderede viewpoints består af et sæt af modeller, der kan bruges til at belyse udvalgte arkitekturrelaterede emner med hensyn til produktionssystemets produktionsevne og designet af det tekniske system.

I forbindelse med beskrivelse af produktionssystemers arkitektur er der behov for at støtte udveksling og behandling af arkitekturoplysninger inden for et bredt sæt af interessentdomæner og værktøjer igennem systemet livscyklus. For at støtte sådanne aktiviteter introduceres der et referencesystem til identifikation af produktionssystemets vigtigste elementer. Referencesystemet udgør en del af rammeværket for produktionsarkitekturbeskrivelser og er baseret på ISO/IEC 81346 standard serien. Systemet giver mulighed for identifikation af systemets vigtigste elementer ved brug af identifikatorer genereret på baggrund af produktionssystemets kompositoriske strukturer.

Nøgleord: produktionssystem, systemarkitektur, arkitekturbeskrivelse, referencesystemet

Preface

This PhD dissertation documents the outcome of a research project conducted at the Technical University of Denmark. The project was initiated in 2009 and the main research activities were concluded in 2013. The project has been interrupted after 14 months due to the loss of the collaborating industry partner (APC). The project was resumed after 4 months, now with production system architecture and platforms as the research subject and Grundfos as the main collaborating industry partner. Consultancy of relevance to the new research subject was carried out in the intervening period.

The research project has been funded by four sources:

- The Technical University of Denmark
- The USEC research consortium
- Grundfos
- APC by Schneider Electric

The target audience of the research is primarily researchers in engineering design, but it is my hope that the dissertation will also find an audience within industry.

My own interest in the subject of architecture, platforms and systems design is what has driven me to take on a PhD study. I feel fortunate to have had the opportunity to engage in research within a subject I feel is so very important for engineering design. The chance to both apply and expand my knowledge in the subject is one I will always be grateful for. With that being said, a PhD study is not an easy undertaking and certainly not one that you can hope to complete without help and support. There are several people I would like to thank in this regard.

First of all I would like to thank my supervisors Professor Niels Henrik Mortensen and Professor Lars Hvam for their guidance and support throughout the project. Their interest and confidence in my research has been invaluable for the completion of this project.

I would like to thank Grundfos and particularly the employees of the Technology Center for providing me with the opportunity to conduct my research under such fine conditions. I hope they have found the experience as fruitful as I have.

A special thank you goes out to my colleagues at the Concept & Sales department in the Grundfos Technology Center for sharing their knowledge and experience, and their invaluable help in carrying out the research. I don’t think I could have asked for a more welcoming and open group of people to work together with throughout this project. I have long since lost count of the insights, ideas and questions they have inspired.

I would also like to thank my colleagues at both the section of Engineering Design and the section of Operations Management for their support, encouragement and collaboration.

I think it is a well-known fact that a PhD project is not without its costs to the PhD student and their family and friends. Even more so when the nature of the research requires travel away from home for extended periods of time. I would like to thank my family and friends for their support throughout this time and look forward to once again being able to say yes more often than no.

To all those who have in any way contributed to this PhD, but have not been mentioned above, I would also like to extend my gratitude for you contributions and support.

And finally my thanks to you, the reader. I hope that you may find the subject at hand as interesting as I do, and that this dissertation is able to inspire your curiosity

Allan Dam Jepsen

Copenhagen, September 2014

Content

Part 1: Setting the stage for modeling production system architecture

Part 2: A contribution to a theory of production system architecture

Part 3: A contribution to production system architecture description

Part 4: A contribution to viewpoints in a Production System Architecture Framework

Part 1: Setting the stage for describing production system architecture

The objective of this research project is to increase the knowledge of architecture-centric development of production systems with a focus on description of production system architecture. The research will contribute with (1) a framework for conceptualizing the phenomenon of production system architecture, (2) a framework for generating production system architecture descriptions, (3) operational modeling methods to support description of production system architecture and (4) a reference designation system to support handling of design information within an architecture description. Part 1 of the dissertation will present the research objective, the scientific and practical goals, the theoretical basis and the research setup.

1 Introduction to the research area

Companies operating in both local and global markets are confronted with demands that are growing as fast as or faster than the companies can fulfil from customers and shareholders alike. At the same time the globalization of businesses and innovations in technology and businesses operations are leading to an ever growing field of competitors, which threaten once secure market positions. To remain competitive and please both customers and shareholders, companies must continually seek to improve their business performance, by employing a wide range of initiatives aimed at improving products, production, organization, processes etc. Production has always been recognized as a one of the most important areas of improvement, because of its vital role in any company´s survival and as a primary source of competitive advantage. Improvements within production have traditionally been aimed at cost, quality and productivity, but as the incremental gains of such improvements become smaller and smaller, the focus is shifting to the role of production as a key enabler of the company´s product strategy and its ability to handle future challenges. Four of the main initiatives that companies are employing in improving production system design are Lean Manufacturing, Six Sigma, Changeable Manufacturing and Architecture & Platform Based Development.

Lean Manufacturing: Lean manufacturing is a production philosophy originated by the Toyota car company, which seeks to balance production resources and value creation by reducing waste as seen from a customer perspective. The term Lean Production System was first introduced by (Krafcik, 1988). (Womack and Jones, 2003) describes the seven commonly agreed upon types of waste: Transport, Inventory, Motion, Waiting, Over-production, Over-processing and Defects. Using Lean manufacturing as a principle for design of production systems can enable reduction of labor, inventory, lead time, capital investments and floor space used.

Six Sigma: Six Sigma was developed by the Motorola company in 1985 as a methodology for quality improvement in manufacturing, based on a quantitative description of the statistical process performance. At its inception Six Sigma was a statistically based quality improvement technique, which could be used to reduce the rate of defects in the production of a product. The goal of Six Sigma is that no more than 3.4 defects, as defined by the customer specification, occur per one million produced products. Since its inception Six Sigma has evolved to become a quality based management strategy pursued by some of the leading companies in their field (Harry and Schroeder, 2000).

Changeable manufacturing: Changeable manufacturing is a collective term for a wide range of design philosophies for production systems that are all concerned with the efficient development, deployment and operation of production systems, which can change according to the need of the company. Drivers of change in production such as new product introductions, new product life cycles, changing and increased product variety, increased outsourcing and a move towards manufacturing networks, require design of production systems that can adapt to the changing production requirements. (Wiendahl et al., 2007) defines five changeability classes by which the design of manufacturing systems can be classified: Changeoverability, Reconfigurability, Flexibility, Transformability and Agility. Two of the best known changeable manufacturing systems are reconfigurable manufacturing systems, and Flexible manufacturing systems. A reconfigurable manufacturing system is one which from its inception is prepared for rapid change of its hardware and software structure, to adapt to changing capacity or functionality requirements. A flexible manufacturing system is a system that can react to changes with flexibility in the manufacturing process and routing of parts.

Platform based development: Platform based development is best known from product development, where it has been proven to offer many advantages over traditional product development, such as offering greater external variety to customers while maintaining lower internal variety within the company; allowing for shorter product launch cycles; and enabling greater productivity in development. Platform based development relies on a cross organizational focus on the underlying design principles, structures and capabilities of a system; and involves the development of subsystems or design entities that can be used in several products. Platform based development offers the possibility of leveraging the beneficial effects of design philosophies such as Lean manufacturing, Six Sigma and Changeable manufacturing across multiple production systems, thereby maximizing the effects of good production system design. At the same time it offers the traditional benefits of platform based design, in reducing development time, freeing up development resources and allowing for technology or design upgrade across the range of production systems. As a consequence of these benefits platform based product development has become one of the more popular means of addressing many of the company challenges and is now increasingly sought as a means of achieving the same effects in the development of production systems.

Focus on architecture in development of production systems

Using design philosophies such as Lean manufacturing, Six Sigma and Changeable Manufacturing as the design guideline for production system design allows for the design of production systems capable of supporting a large part of a company´s strategy, but it does not necessarily support the successive design of multiple production systems across the product portfolio; across different generations of production systems; and throughout a company´s entire production systems. This is evident for many design philosophies associated with changeable manufacturing that focus on adaptation to a changing product portfolio. Such philosophies bare a resemblance to platform based design, but rarely do they take advantage of or focus on the successive development of multiple production systems. Instead they focus on the changeability of single production systems.

Platform based development does offer a means of leveraging other design philosophies for design of multiple systems, but it has also yet to be fully embraced within production system design despite the proven advantages within product design. Production systems are still to a large extent custom made for each new product that must be produced, and the degree of reuse of designs and systems across products, product generations, and production sites is often limited to the equipment supplier side of the equation.

In order to better leverage established design philosophies; to strengthen the adoption of platform based development of production systems; and to support a focus on the production system as a key enabler for company strategy and tactics in addition to operations; it is important that the production system architecture be the center of attention in the development of new production systems. This means that the architecture must be the prevailing theme in analysis and design of the production system. Architecture-centric design is already a key element in many design philosophies, including but not limited to the ones mentioned here, and it offers the possibility of greater leveraging of design solutions across multiple production systems. However leveraging of such design philosophies for multi-system development will require a greater understanding of the role of production system architecture in relation to the company´s various doctrines, strategies, tactics and operations. Architecture-centric production system design should enable strategic thinking and long term development of production systems across multiple generations of production systems and applications to address the future challenges of companies. In order to achieve this, a greater understanding is needed of the theoretical and operational aspects of architecture-centric production system design.

This research project expands upon the understanding of the architecture phenomenon in the context of production system design, and seeks to provide better support for architecture-centric production system design through description of the architecture and support for handling of architecture related design information. The primary focus is on production system architecture, and the aim is to describe a theoretical framework for conceptualizing production system architecture, followed by the development of operational tools for modeling and handling production system architecture. The research will be based on theoretical and practical results from the product design field, to leverage existing knowledge from this field, where the architecture-centric design philosophy is considered most mature, as it is an integral part of platform based product development.

2 Research justification

The rising challenges for companies relying on industrial production in an increasingly competitive world, is the subject of interest for both the government, industry and the scientific community. Governments see a threat to employment and the national economy, and the industry sector is well aware of the challenges to overcome to remain competitive and profitable. Both within governments and industry a very large number of means are debated and applied as possible solutions. Many of these fields are in need of further study, so it seems only relevant to justify a research focus on architecture-centric production system design. What follows is a review of some of the main reasons behind the chosen research subject.

2.1 Response to business challenges

In 2004 a joint European National/Regional Technological Platform (NRTP) called “Manufuture” was established with the goal of connecting manufacturing stakeholders and influencing the direction and goals of future European research activities in the manufacturing field by industry and the scientific community.

The objective of the Manufuture NRTP is to ensure (Manufuture-EU, 2013):

- Competitiveness in manufacturing industries
- Leadership in manufacturing technologies
- Eco-efficient products and manufacturing
- Leadership in products and processes, as well as in cultural, ethical and social values

A national initiative under Manufuture, known as “Manufuture-DK”, was established in Denmark, and in 2009 a joint project called “Manufacturing 2025” was carried out within Manufuture-DK. Five leading Danish manufacturing companies, the two largest labor market organizations and researchers from three Danish Universities participated. The purpose of the project was to ascertain the conditions and challenges facing manufacturing companies if Denmark and other European countries are to maintain a strong and competitive manufacturing sector; and to inspire research and development directions for manufacturing in Denmark (Manufuture-DK, 2013). Manufacturing 2025 offers an insight into some of the challenges faced by Danish and European production companies (Johansen, Madsen, Jensen and Vestergaard, 2010) and further investigation at Grundfos A/S has resulted in the identification of the following challenges as justifications for researching architecture-centric design of production systems, as listed below and elaborated upon:

- Cost pressure
- Global value chains
- Innovation & technology development
- Product variety
- Time to market
- Sustainability
- Offshoring and outsourcing of manufacturing knowledge

A catalogue of proposed research subjects has been made based on the manufacturing 2025 project (AAU, 2010). Among the seven themes is research into “modular platforms”. Both platforms in themselves and modular design in general rely on an architecture-centric design approach, and as such where ever platforms or modularity are of interest so is also the architecture. This focus on architecture and platforms can be collectively referred to as architecture & platform based design. Below is described how platforms and architecture relate to each of the listed challenges for businesses:

Cost pressure: The cost level in many countries is increasing and it has for some time now forced a shift of traditional production to low cost countries. For example, Denmark´s average rank for the eleven indicators for cost within the OECD countries is number 32, as a consequence of high salaries and taxes (DI, 2013). This places Denmark towards the bottom of the list, and the high cost level increases the risk of companies moving production and investments abroad, or that foreign companies will disregard Denmark as a place of operation. Keeping production in high cost countries typically focuses on reducing the operating cost of production systems, however lowering and/or postponing capital investments is equally important. While operator salaries are high, lowering and/or postponing capital investments for production systems can quickly compensate for the salary level, thereby allowing for production to be maintained close to customers or development centers. If the economies of scale that platforms have been seen to produce for products is also achievable for production systems then platforms may well be a way of lowering or postponing the resource cost and capital investment needed for the design, build and procurement of production systems. Additionally the modular design which is also applied in many designs may very well allow for gradual capacity increase for production systems, thereby postponing the need for investment in equipment before the production capacity is needed.

Global value chains: Production for many companies is going global either to achieve cost savings or to move production closer to markets. As a result production system can very easily see a great variety in solutions across the global production. As products mature or capacity requirements change, some companies may also transfer production systems to other production sites. This means that the location of a production system may change throughout its life cycle. Utilizing the same underlying architecture for the design of production systems, which may be deployed across different production sites, means that to a large extent the production systems can be treated the same in operation, service and maintenance, making transfer and subsequent run-in much easier, since the systems will be recognizable to the global production organization.

Innovation: The competition from low cost countries is no longer relegated to low tech production technologies. As once unique production technology becomes available to more and more companies, companies must increasingly innovate and develop new technology. While the global financial crisis of 2008 drove many companies to reduce cost by drastic measures, many companies recognized that staying competitive required them to maintain or increase their research budgets. Production today must be prepared for more frequent technology change, and be prepared for upgrading the technology of existing production systems. Ensuring that this can be done efficiently, quickly and across many production systems within the company, requires a commonality of design which is in line with the purpose of architecture & platform based design.

Technology development: As technologies mature companies will also push for greater innovation and new development. The push is to prioritize development resources for new innovation rather than continued design based on old/known technologies. Keeping a common architecture for multiple production systems offers one way of freeing up resources for new development and innovation, by reducing the design task for successive production systems based on the same architecture. As a company develops new technology or higher performing design solutions, there is an impetus to leverage the technology across as large a part of the production as possible. Incorporating a new technology in a platform design offers a planned means of deploying technology for several production systems to be developed. A commonality of the underlying architecture will be needed as an enabler.

Product variety: There is a drive for companies to offer greater product variety, and one of the means of doing so is product platforms. Product platforms allow for greater external variety for customers with lower internal variety in the company. The sharing of design across products and product families carries with it a commonality of production, which is ideal for the sharing of a common architecture in the design of production systems. With greater sharing across the production output the potential reuse for production solutions increases. The sharing of design across products and product families results in a commonality of production, which can carry over into the design of multiple production systems for different product families or production locations. With greater use of platform based design for products thus follows the potential for greater use of architecture & platform based design of the production systems, since in a manner of speaking the “market” for the production systems based on a common platform increases.

Time to market: It is a common theme for most companies that new product introductions are required to happen more quickly and more frequently. The push for faster time to market, and the increase in new developments, is driving a need for the design and build of production systems to be faster and require less design resources. Architecture & platform based design offers one way of keeping up with the increased pace and volume of product development, by reducing the design task, and potentially allowing for earlier build or procurement of equipment, and making run-in easier through well-known design.

Sustainability: In recent years the focus on sustainability and the resource foot print of production has increased dramatically. Companies are seeking to make their production more environmentally friendly both by choice, customer demand and regulatory demand. Platforms offer a way of encapsulating the sustainable design and leveraging a particularly well performing design for the design of as many production systems as possible based on a common architecture. At the same time, developing a shared design can justify the use of a larger development effort to achieve a more environmentally friendly design, when the design will be used for more than one production system. Employing a common architecture that is focused on sustainability can thus support the leveraging of sustainable designs through platforms used in multiple systems.

Offshoring and outsourcing of manufacturing knowledge: For companies having production in high cost countries, keeping production in these countries presents a challenge for the company´s profitability, and so many companies have turned to offshoring and outsourcing as a means of reducing costs. A great deal of the debate of offshoring and outsourcing has focused on the loss of jobs in the companies and/or countries from which the production is moved. However, as more companies have gained experience with offshoring and greater outsourcing, it has become clear that the movement of production very much needs to be considered not just in relation to a company´s finances but also very importantly in relation to the company´s knowledge base. A blind focus on the financial aspects of giving up or moving production, ignores the key role production plays in the generation of knowledge for a company. When offshoring or outsourcing production, crucial input for the development activities of a company, particularly product and production R&D, is either diminished, hindered or lost entirely despite the plethora of communication options available today. The same is true of outsourcing, if not more so. This realization has resulted in a counter movement of reshoring and insourcing of production to maintain production as a knowledge generating force in companies. Keeping production close to the R&D organizations of the company is therefore increasingly seen as important for companies relying on new developments for a competitive advantage. Architecture & platform based design can be an enabler of this counter movement if it is used to design production systems for use in very different production settings e.g. countries with high or low labor costs; or lower the production system cost. The design and documentation tools used in architecture & platform based design methodologies can also offer a means of documenting and communicating the production knowledge.

2.2 Response to an academic interest

After years of research into the subject there exists a broad body of work within the field of architecture & platform based product development, but increasingly there is also a focus on the production related architecture and platform aspects, not just by the nature of the dispositional relations between product and production, but by virtue of the need for design of production systems under the same time and cost constraints of product development. The contributions span theories, tools and methods within among other:

- Product architecture and platforms
- Production system architecture
- Process platforms
- Alignment of product and production architecture
- Technology platforms
- Production system platforms
- Operational modeling tools for architecture and platforms

Product architecture and platforms

(Jiao, Simpson and Siddique, 2007) demonstrates the broadening of research within platform based design through a review of a broad selection of platform related literature. Research into product architecture and platforms now covers a broad field of subject matters spanning from business strategy (Meyer and Lehnerd, 1997; Pine II, 1993) to architectural design (Harlou, 2006).

Production system architecture

A great deal of research has always been concerned with the architecture of production systems, spanning cost and time optimization for example; through Group Technology as early as the 1920´s; and on to current research themes within changeable manufacturing such as transferability, flexibility and reconfigurability of production systems (Wiendahl et al., 2007).

Process platforms

Taking its start in a recognition of the need for aligning product and production design, the area of process platforms has received increased attention since the turn of the century, and there is now a focus, not only on the alignment between product and process architecture, but also in the need to focus on the development of platforms in production in general across products and product families (Sanchez, 2004; Schierholt, 2001; Jiao, Tseng, Ma and Zou, 2000; Jiao, Zhang and Pokharel, 2003; Berglund, Bergsjö, Högman and Khadke, 2008; Högman, 2011).

Alignment of product and production

Authors such as (Olesen, 1992; Sawhney, 1998; Andreasen, Mortensen and Harlou, 2004; Mortensen, Pedersen, Kvist and Hvam, 2008a) stress the importance and advantages of aligning product and production design. And many authors have explored the connection between product and production in variety creation (Salvador, Forza and Rungtusanatham, 2002).

Technology platforms

Technology platforms are now being considered in a context between products, production systems, product platforms and process/production platforms (Högman, 2011; Berglund et al., 2008; Levandowski et al., 2012) as further understanding is sought of the relations between technology, product design and production design.

Production system platform

The possibility of applying platform modeling tools from product development within the domain of production platforms is now being investigated, to see the possible applications, differences and similarities (Pedersen, 2010).

Operational product architecture and platform modeling

Operational modeling tools are available for architecture & platform based product development and are the subject of ongoing research particularly within the Product Architecture Group at the Department of Mechanical Engineering at the Technical University of Denmark (Harlou, 2006; Andreasen, Hansen and Mortensen, 1996; Mortensen et al., 2008b; Kvist, 2010; Pedersen, 2010).

As is clear the phenomena of architecture and platforms for products and production systems are already the subject of research and have been for some years. This research projects seeks to expand the existing body of knowledge with further understanding of architecture and platform phenomena within production system design and to develop operational modeling tools for the support of architecture & platform based production system design.

2.3 Interpreting the needs

Architecture-centric design of production systems can be one way of addressing many of the challenges faced by businesses. Utilizing the design philosophy can help in attaining these effects:

- Lower capital investments through economies of scale in design, build and procurement
- Postpone capital investments through modular architecture and platform design
- Enable easier run-in, service and maintenance when transferring production systems to new production sites
- Deploy developed production technology through platforms used in production systems sharing the same architecture
- Ensure use of preferred design solutions or solutions which show the best characteristics in sustainability, quality, productivity, safety etc.
- Increase productivity in the design of production systems and free up resources for innovation
- Faster design, procurement, installation and run-in of production systems
- Enabling easier technology upgrade across the range of productions systems in a company
- Lower time to market through design reuse

Many companies are now seeing the potential of using platforms to keep pace with market demands and the pressures of low cost production in other countries. An architecture-centric design approach is very important in such platform based development. As companies seek to adopt practices of using production platforms they discover the need for...

- …a vocabulary for production architectures and platforms
- …an understanding of how architecture and platforms relate to the technology and engineering design of production systems
- …uncovering what the differences and similarities are between the applications of architecture and platforms of production systems built in low volume versus the present use of architecture and platforms for design of products manufactured in comparably high volume.
- …defining how architectures and platforms can be adopted for design of production. The development of new production systems may be prepared by developing a set of modules with well-defined interfaces to the rest of the production system.
- …changing production systems from being individual systems developed for an individual product family, to being systems consisting of designs and modules that can be used across many projects. The product development process changes in the same manner, namely from developing single production systems to developing multiple designs that can be used in many situations.
- …how to handle the development of production system architecture and platforms in an operational manner

While the scientific community is looking at architecture for both products and production systems and the relations between them, much of the research in production system design is centered on the task of designing single production systems, and aligning the product and production system architecture. If architecture-centric design of production systems is to be the subject of the same planning and development strategies as product design, then there is a need for greater understanding of the architecture phenomena in production system design. Production system design must to a greater extent be approached as a study in the design of multiple production systems, rather than as the discipline of designing an optimal architecture for single production systems intended for the production of specific products. There is a greater need for viewing the design of multiple production systems in its own right, and to investigate how architecture-centric design of production systems can be utilized across products, product families, production sites, generations of production systems etc. Given the proposals for research subjects based on the Manufacturing 2025 project, and the challenges described, the author finds the research subject more than justified.

2.4 Scope of this research

This research project expands upon the understanding of the architecture phenomenon in the context of production system design, and seeks to provide better support for architecture-centric production system design through description of the architecture and support for handling of architecture related design information. The research rests on three basic assumptions.

The first assumption relates to the complex nature of the design process for production systems, and the need for making decisions in an environment where many different stakeholders are involved and design requirements span a multitude of subjects.

The second assumption relates to the connection and shared theoretical basis of design of products and design of production systems, which is used as the basis for much of the vocabulary definitions and modeling tool development in the research project.

The third assumption relates to the need for dissemination of design tools proposed by scientific research and the need for handling of architecture and platform objects in a very large and diverse group of IT systems and design tools.

Assumption #1: Assumption on explicit and visual models

"Explicit and visual models of production system architecture enable better decision making regarding development of production systems"

Developing architectures for production systems requires handling of a wide range of requirements by many different stakeholders involved with the design of both the architecture and subsequent production systems based on the architecture. The assumption states that explicit modeling by visual means enhances the stakeholders' ability to absorb and process complex design problems, and allow for easier communication between stakeholders for better decision making. The benefits of using visual modeling to facilitate communications among different stakeholders has been demonstrated previously by (Alabastro et al., 1995), and throughout published literature from the Product Architecture Group at the Department of Mechanical Engineering at the Technical University of Denmark.

Assumption #2: Assumption on relation to product design

"The architecture phenomena for products and production systems share a conceptual basis, but they are still distinctly different because of the differences in the design context and life-cycle of products and production systems."

Production systems can be viewed as products in their own right, although they are products intended for operation within the company and not be the customers of a company. The second assumption recognizes this perspective on production systems, and makes the assumption that there will be a sharing of concepts between product and production systems in regards to the architecture phenomenon, but that these will be separated among other by the context of the life cycle for the production systems being different. Production systems exist as the means of producing products, and they are subject to requirements internal to the company.

Assumption #3: Assumption on communication of architecture elements

"Explicit coding and referencing of design objects enhance and support the use of multiple different design tools and information systems for architecture & platform based design of production systems."

The assumption is based on an observation that there are a large number of design tools and IT systems involved in the design of production systems. The assumption states that the effectiveness of using all these systems is to a large extent dependent on the ability to identify and reference design objects of architecture within and across the tools and systems. This assumption is part of the justification for developing an information handling support tool, as detailed in Part 5 of the dissertation.

3 Structure of dissertation

This dissertation is structured as follows.

Part 1 - Setting the stage for modeling production architecture

Part 1 provides the setting and background for the research, and gives an introduction to the research problem, objectives, approach and setup.

Part 2 –A Contribution to a theory of production system architecture

Part 2 investigates the architecture phenomenon for production system design and contributes to the existing vocabulary within architecture & platform related design theory. A conceptual framework for the architecture phenomenon is developed, and it is sought to relate the production system architecture phenomenon to definitions from related fields of research and industrial applications.

Part 3 – A contribution to description of production system architecture

Part 3 provides a conceptual model for architecture descriptions and investigates the necessary elements of architecture descriptions for production systems. The conceptual model has two purposes: 1) to provide a general framework around which to relate scientific contributions to the description of architecture of production systems and products, and 2) to provide a specific framework around which to relate the modeling and information handling contributions of this particular research project. Part 3 also presents the requirements for the use and content of a production system architecture description defined in relation to the conceptual model.

Part 4 – A contribution to architecture description viewpoints and model kinds

Part 4 presents a contribution to a reference framework that can form the basis of generating architecture descriptions for production systems. The so-called Production System Architecture Framework (PSAF) provides two viewpoints on the production system, with the possibility of adding more viewpoints to address other stakeholder concerns. The viewpoints consist of model kinds that help system architects and other stakeholders involved in the architecting process address specific architecture related concerns.

Part 5 –A contribution to correspondence in a Production System Architecture Framework

Part 5 presents an object coding tool that enables correspondence between the models of an architecture description. The tool is intended to support handling of information regarding the key constituent elements of a production system by coding and referencing of architecture design objects and structures across stakeholder domains and design tools. The tool is related to existing industrial standards for object identification and referencing, among them the ISO/IEC 81346 standard series.

Part 6 – Conclusions

Part 6 concludes the research and discusses the possible impact of the research.

4 Scientific approach

This section describes the scientific approach of this research project this includes the phenomena to be studied, the research objectives and questions, and the research setup.

4.1 Phenomena to be studied

Three phenomena related to production system design will be studied in this research as described below i.e. architecture, architecture description and architecture information handling.

Architecture: The study of the architecture phenomenon aims to quantify what can comprise architecture for production systems. What are the similarities and differences to other architecture concepts such as product architecture and system architecture, and what relevant sub-phenomena exist in relation to the different levels or kinds of production systems and the systems life-cycle? The phenomenon will be explored in the context of designing multiple production systems sharing characteristics of design or capability which may be developed in parallel or sequence. The focus is not on sharing of development resources, but on similarity of design. The aim is to study the phenomenon as a combined development task within the company and how it relates to the architecture of multiple systems and their intended applications.

Architecture description: This phenomenon is related to the description of architecture in a design context. Describing the architecture of one or more production systems is considered a key element of the design process, and as such it is relevant to explore how to support the design process through description of the architecture. The research will cover both the requirements for architecture description in a multi-stakeholder environment and the possible means of description.

Architecture information handling: The last phenomenon to examine is the treatment and exchange of architecture information. Many different stakeholders, tools and methods are involved in production system design, so information about the architecture must be treated by and exchanged between many different stakeholders and systems. To effectively handle architectures in this context, an understanding of what information should be exchanged and how this can be supported is needed.

4.2 Theoretical research objectives

Theories of architectures and platforms exist, but are most often related to product design. Theories related to design of production systems architectures are also being strengthened in the scientific community, but are often focused on the design and capabilities of single production systems, or do not address the production system design in the context of the system’s different roles in entire company.

The theoretical contributions of this research project are intended to heighten the understanding of the architecture phenomenon for production systems, and relate it to existing theories within production and product design. Following in the footsteps of previous work within modeling and information handling for architecture and platform based product design, operational tools will also be developed and tested to aid in the practical application of architecture-centric design of production systems. There are four main contributions within this research:

Understanding of the production system architecture phenomenon: This research will contribute to the understanding of production system architecture in the context of design of multiple production systems over time, for the use in multiple production locations, and for production of multiple products. A conceptual framework is proposed which represents a means of understanding the architecture phenomenon and its different sub-phenomena. The framework expands on the understanding of the architecture phenomenon and establishes the vocabulary necessary to relate sub-phenomena of architecture. The framework can also serve as an aid for companies who wish to move towards architecture-centric or platform based design of their production systems, or serve to provide a structured understanding of architecture for companies that already make use of a certain degree of reuse in design across production systems.

Conceptual model for architecture descriptions: This research will contribute to the description of production system architecture by specifying the content of production system architecture descriptions. Such architecture descriptions are aimed at supporting cross stakeholder communication, analysis and design of architectures, and documentation of key architecture design decisions. A conceptual model for architecture descriptions is developed based on ISO/IEC/IEEE 42010, and the relevant stakeholders and stakeholder concerns to be addressed in the architecture descriptions of production systems are investigated. The conceptual model describes the content and conceptual relations of architecture descriptions that describe aspects of the production system design of concern for key stakeholders.

Viewpoints and model kinds for architecture description: A large part of this research is aimed at providing operational support for the description of production system architecture. As part of this goal a contribution will be made in the form of two architecture viewpoints and their associated model kinds. These architecture viewpoints are defined as part of a reference framework for production system architecture descriptions that can be used as the basis for architecture descriptions of different production systems. The model kinds in the PSAF are partially based on known modeling tools from architecture & platform based product design that have been adapted for use for production systems; along with new model kinds. The modeling kinds rely on visual modeling of architecture design objects and requirements, and are meant to support the principle production system design rather than detailed engineering design.

Reference Designation System for architecture information handling: The fourth major contribution focuses both on how to document the key constituent elements and relations of production system architecture; how to efficiently communicate this information in the production system life cycle; and how to interconnect the models included in an architecture description. The contribution consists of a coding tool, a so called *Reference Designation System*, which can be used to express key aspects of the constituent design of a production system. The developed reference designation system can serve as a correspondence kind supporting the myriad communication needs in architecture-centric production system design. The system can among other things tie together different design tools such as DSM and Interface diagrams, and act as a basis for structuring design objects in data systems such as PDM systems. The system is based on the ISO/IEC 81346 standard series.

4.3 Practical research objectives

The practical research objectives are related to the execution of the research project, the contribution to industry, the relation between research results and industrial methodologies, and consolidation of research conducted by the research group of which I am a part. There are four main objectives:

Execution of research

It is the practical goal of this research project to conduct research directly in industrial settings where observation and experiments can occur. Direct observation in industry will provide a crucial input for the understanding of the production system architecture phenomenon, and will be the basis of developing modeling and information handling tools.

Contribution to industry

It is the general objective to support industry professionals in the description of production system architecture and their understanding of the production system architecture phenomenon. This is to be realized both by providing an explanation of the architecture phenomenon and by providing operational tools for modeling of architecture and referencing of architecture elements. Dissemination of architecture and platform related theory to industrial partners through seminars, presentations and development projects, is also intended to spread the knowledge of developments within architecture and platform theory from academia to industry in Denmark.

Specifically this research project will also aid the main industrial research partner in formulating and initiating a shift towards architecture & platform based design of production systems and heighten the existing degree of design reuse in the organization. This is realized both by engaging in the formulation of strategy concerning architecture and platforms, and by providing tools and methods for project practitioners.

Bridging gap between academic research and industry

Much of the research conducted in academia faces a great hurdle in bridging the gap between academia and industry. One of the main practical goals of this research is to aid in the industrial dissemination of research results within the field of architecture-centric design. It is sought to do this by seeking inspiration in industrial concepts, methodologies and standards when formulating theory and when developing the contribution of architecture description. Specifically the research project will be related to industrial standards of relevance to the research subject.

Means of consolidating research results from research group

The research group of which I am a part has for a long time conducted research into architecture-centric and platform based design of products and systems in general. The results of this research include a collection of modeling tools that have been interrelated to some degree. It is my observation that there is a potential for greater consolidation of the research results of the group, and that there is a need for a general framework in which this can occur. Part of the practical goals of this research is therefore also to try and provide an input for the consolidation of modeling related research results from the research group.

4.4 Research questions

Three research questions form the base of this research.

Research question 1

Research question 1 is based on the desire to understand the phenomenon of architecture for production systems and how it relates to other uses of the concept of architecture in particular as the concept is understood within the field of product design. The question is intended to provide a theoretical framework around which the phenomenon of production system architecture can be approached. It should allow for further investigation of individual elements of the concept, and identify relations of interest between production system architecture and other concepts.

It would be easy to simply adopt the architecture concept as it is understood within product design. Indeed one way of looking at the design of production systems, is to consider the production system as a product in itself, although one intended for use within the company. Such a product would be characterized by…

- Low volume or engineer-to-order production
- One-of-a-kind design or low volume design reuse
- Shifting requirements due to the design and build of individual production system being separated by time, sometimes several years. Many requirements would change in the intervening time, for example due to:
    - The product(s) to be produced changes
    - New technology becomes available
    - Suppliers phase out products
- Broad specification ranges for shared architectures, in order to cover vastly different requirements. The span of requirements would be wide for requirements related to…:
    - Production location (cost, regulatory, sourcing)
    - Products to produce
    - Capacity needs
    - Changeability needs

The reason for not outright considering production systems as products is primarily due to the special position they hold in the product life cycle and the role they play in a company. Production systems also represent a class of systems that are the subject of their own design disciplines with specialized tools and methods. It is therefore reasonable to question whether or not the same architecture definitions and phenomena for products hold true for production systems, or if not, what are then the characteristics of production system architecture? This need to better understand production system architecture is what lies behind the first research question.

> ***RQ 1*** *What is production system architecture and how does the concept relate to existing theories of architecture within design of products, systems, and production systems?*
>
> <u>Supporting questions</u>
>
> - *What phenomena are described by production system architecture?*

- *How does production system architecture relate to existing concepts of architecture and platforms in product design, systems engineering and production system design?*
- *What levels of production system architecture can be defined?*

Research in product design, has shown that there is no single commonly agreed upon definition of what constitutes a product architecture or product platform. It would therefore be foolish to try and directly apply definitions from the product domain to the production domain, without further investigation of the concepts. However the aim is not simply to try and describe production system architecture as a separate concept, but to identify how it differs from existing concepts. The research question is intended to investigate the constituent design and related phenomena of production system architecture, and to frame the concept in the context of other kinds of architecture or platforms. The research question should lead to a reference framework for the architecture phenomenon, which can potentially be used for several different things such as:

- Identifying architecture types and levels in a company
- Articulating a strategy of architecture-centric and/or platform based development
- Setting up an organization for ownership, development and use of architectures

Research question 2

The development of a production system based on a focus on the architecture can involve a very large number of stakeholders from many different domains. If the architecture is to be easily communicated between these stakeholders, there is a need for effective describing the architecture and the requirements which form the basis of decision making. The second research question is the starting point for figuring out how the architecture can be modeled for so that it can be understood and treated by different stakeholders.

> ***RQ 2*** *How can production system architecture be described, and what are the relevant elements and phenomena to describe in order to best support decision making on the design by stakeholders from different disciplines?*
>
> <u>Supporting questions</u>

- *What stakeholders and stakeholder concerns should be addressed by a production system architecture description?*
- *What constituent elements and relations of a production system should be modeled as part of a production system architecture description?*
- *How can the architecture of a production system be modeled visually, including phenomena of scalability, interchangeability and flexibility of production systems be modeled?*

Research in design of product architecture has shown the value of modeling phenomena and requirements in a visual manner to support the design process. Describing key elements and phenomena for production systems to make decisions on scalability, changeability, interchangeability of technology etc. is similarly believed to support good decision making in the design of production systems by facilitating communication of the system capabilities to relevant stakeholders e.g. when stakeholders from product development and production development discuss product and production design requirements. In order to do this, it is necessary to identify the relevant aspects of the production system to communicate, and to develop an architecture description which provides a sufficient means of communication. A description of the production system architecture based on visual modeling is suggested as one such means, and the research into description of production system architectures will focus on visual modeling tools for this very reason.

Research question 3

Experience in architecture & platform based product design has shown the need for handling of architecture and platforms across multiple stakeholder domains, business processes, design tools, documents and IT-systems. With the advent of engineering methodologies such as concurrent engineering this task has only increased, as the need for efficient communication of the architecture or platform between stakeholders and systems grows. To efficiently execute architecture & platform based development within a concurrent engineering process, and continuously handle the architecture throughout the production system life cycle, there is a need for an operational communication tool, which allows for modeling and referencing artefacts and structures across design disciplines, design tools and IT systems, for all relevant stakeholders throughout the life cycle. This is the motivation behind the third research question.

> ***RQ 3*** *How can the ISO/IEC 81346 standard series be applied to support exchange and processing of architecture information within and between stakeholder domains and tool in the production system life cycle as part of the description of production system architecture?*
>
> <u>Supporting questions</u>
>
> - *What production system elements and structures are relevant to model to increase communication of elements, structures and phenomena between stakeholders, design tools and IT systems as part of the description of system architecture?*
> - *How can the ISO/IEC 81346 standard series be applied for modeling relations between system elements such as type commonality and functional allocation?*

The Product Architecture Group at the Department of Mechanical Engineering at the Technical University of Denmark has always had a focus on both scientific contributions and development of operational tools to support the adoption of research results in industry. As more and more design methods are proposed in academia, and companies adopt new tools, the need for handling architecture related information in different design methods and tools of different nature grows. I believe that there is a need for operational support of architecture handling, and will seek a contribution in this area based on adaptation of existing industrial standards for developing reference designation systems. Using the ISO/IEC 81346 standard series for referencing of equipment is the standard option for fulfilling parts of the European Machinery Directive (EN 60204-1), unless another system is agreed upon between supplier and customer. As such the system is widely applied within the European Union and will be recognizable to many industrial users if it serves as a starting point for a system for exchanging architecture related information. This research projects seeks to adapt and apply the standard series for this purpose as a part of the description of production system architecture.

4.5 Research methods

This research project follows the research paradigm described by (Jørgensen, 1992) (see Figure 1), which states that research is both *Problem based* and *Theory based*, or in other words that the initiator of research can be both a practical problem and a gap in existing scientific theory. The research conducted here is both problem based and theory based, often with new research subjects feeding of each other e.g. new scientific acknowledgement based in a theory base leading to the discovery of a practical problem which initiates new research.

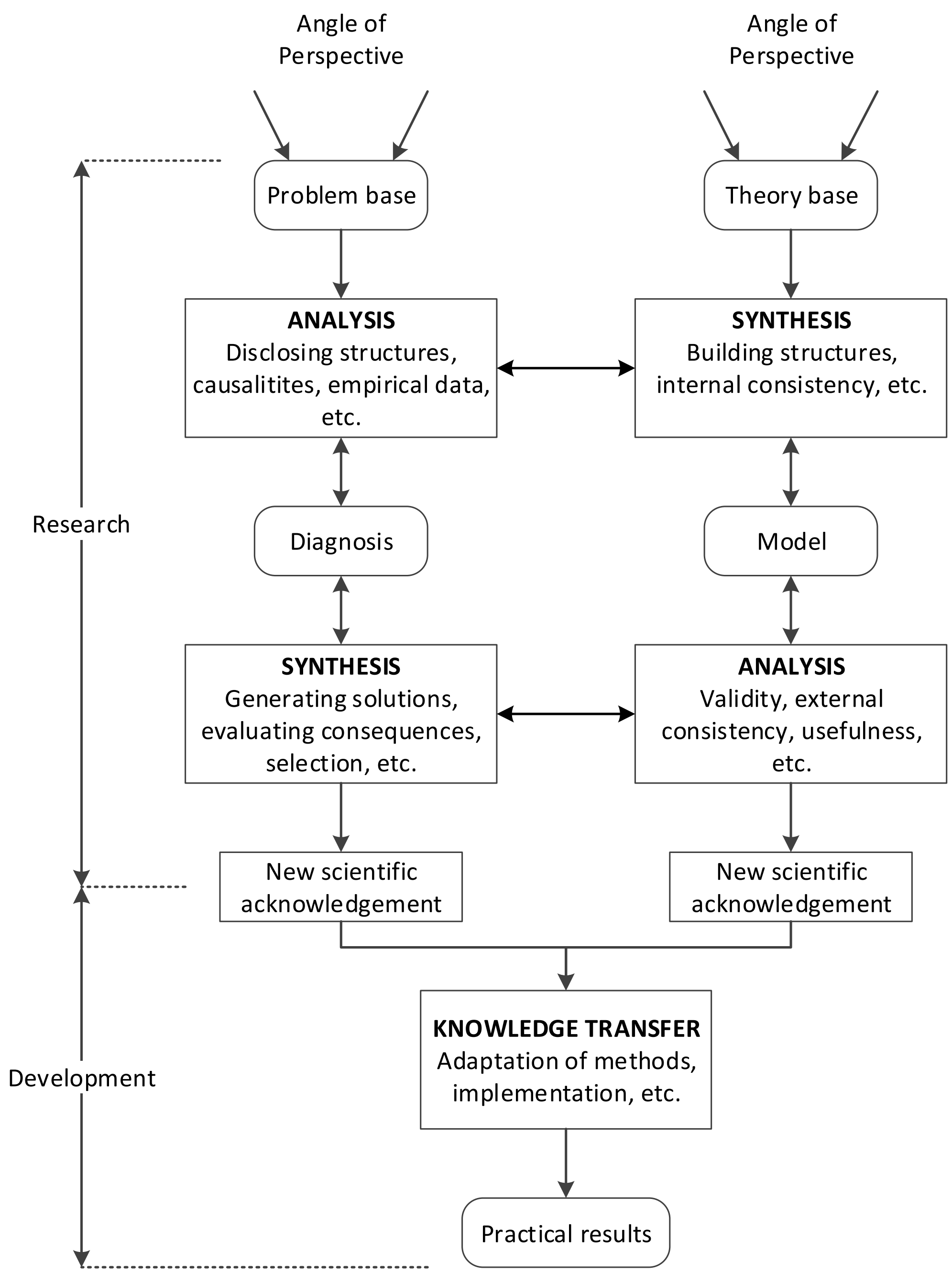


Figure 1 - Basic work paradigm for research and development activities, which recognizes the interaction between theory and practice in scientific research. Figure translated from (Jørgensen, 1992, p.3)

The main initiator of the research project is a recognized gap in the scientific theory that does not adequately describe production system architecture from a theoretical standpoint. Subsequent research activities regarding the support of architecture-centric production system design has grown from a discovery of the need for understanding and solving practical problems of design, and gaining new scientific insights of how to explicitly describe architecture elements by use of visual models and communicate constitutive elements and structures in a production system life cycle.

Scientific philosophy

This research is primarily based on the philosophy of Critical rationalism, where models and methods are improved through means such as observation, experiment, literature study etc., to obtain a better description of an objective empirical reality. Much of the research carried out by the Product Architecture Group at DTU relies on action research, in which the researcher is an active participant, and attempts to both observe and influence the design process. This is also the case with this research. The reason for choosing action research is that the models of this research are intended for use in a context of human action, and it is necessary to subject the models to real world conditions with participation of people from industry as part of the development of the models.

One very important thing regarding critical rationalism should be noted. Critical rationalism relies on falsification rather than verification as the basis of new scientific knowledge, which can be a problem for research of this nature, because we must be able to specify under which conditions the models of this research hold true. The models in this research are developed and tested in a company setting, with conditions that are not always controllable, fully known, unchanging or reproducible. The effects and effectiveness of the models are dependent upon the conditions for the test, for example the available resources (people, time, data) and the focus/prioritization of the projects, system complexity, organization, etc. Not all these things can be quantified, so finding a condition under which the claims of effect and effectiveness can be falsified becomes easy. Human perception is also involved in interpreting visual models, which means that falsification can become a matter of human perception and motivation. Because the research subject is to some extent reliant on human perception, and because action research is the chosen method, the susceptibility to falsification must be carefully considered throughout the research project. In the opinion of this researcher, falsification becomes too easy, when the conditions under which something must hold true, are not fully known, which can be the case in industrial research, because there are so many factors involved. Either the necessary conditions are described so detailed that it becomes impossible to find two testing scenarios fulfilling the same conditions (there will always be a difference in the company, project, project team or production system); or the conditions are specified so broadly, that it then becomes easy to find a condition under which the models do not hold true. To some extent I therefore subscribe to the scientific approach of Critical realism for its distinction between research in social phenomena (which modeling of architecture is to some extent) and natural; it´s embrace of pluralistic methodologies; and the focus on uncertainty of generalized knowledge (Buch-Hansen and Nielsen, 2005).

In taking a general stand of critical rationalism I will strive towards controlling the conditions under which tests are conducted, to explore the limitations of the results. From the approach of critical realism I acknowledge that there is a fundamental restriction in the possibility for testing by falsification, and will seek to generate as good a foundation as possible through a pluralistic methodology using many different research tools. The main tools used in this research project are literature study, empirical observations, semi-structured interviews and experiments by means of participation in development projects of the main case company.

4.6 Research activities

Many different activities make up the body of work in this research project. An emphasis has been made on establishing a contact to industry to ensure that the contributions of the research are based on real world observations and testing of the developed models. Some of the main activities are listed below.

Literature

The study of existing literature has focused on among other: architecture and platforms for products, production and technology; changeable manufacturing; variety creation; product and production alignment; product structuring and coding; industrial standards for system description and information handling; design theory, etc.

Key conferences & research exchanges:

- NordDesign 2010, Sweden, Gothenburg, 2010
- ICED 2011, Denmark, Lyngby, 2011
- Produktudviklingsdagen 2011, Denmark, Lyngby, 2011
- Binational scientific exchange on product family development, Germany, Hamburg, 2012
- 2nd research colloquium on product architecture design, Germany, Munich, 2012
- Open hearing workshop on CCS classification and identification of building elements, Denmark, Lyngby, 2012
- Hearing seminar, CCS classification and identification of building elements, Denmark, Copenhagen, 2013

Courses

- Research and PhD-studies at DTU Management (2,5 ECTS), Technical University of Denmark (DTU), Denmark, Lyngby, 2009
- EDEN Doctoral seminar on research methodology in Operations management (4 ECTS), European Institute for Advanced Studies in Management (EIASM), Belgium, Brussels, 2010
- Getting my research into journals (1,5 ECTS), Copenhagen Business School (CBS), Denmark, Copenhagen, 2010
- Intelligent computer systems for design automation (7,5 ECTS), Jönköping University, Sweden, Jönköping, 2010
- Systems Engineering (10 ECTS), Technical University of Denmark (DTU), Denmark, Lyngby, 2012

Teaching

Teaching within academia: Part of the PhD study has involved teaching engineering students within mass customization, product modeling and system configuration. A part of this PhD study has also included supervising master thesis students. The supervision has been carried out in collaboration with Professor Lars Hvam the co-supervisor of the PhD-project.

<u>Teaching within the companies:</u> Results from this research have been taught in one course and three seminars within Grundfos and American Power Conversion.

- 1 X Platform modeling course at American Power Conversion (15 participants)
- 3 X Platform seminars at Grundfos (250 participants in total)

Experiments at Grundfos A/S

The majority of the research activities have taken place within Grundfos A/S, by means of participation in a strategic initiative for platforms in production. The initiative was newly started when the PhD study began, and in total the research has spanned 2.5 years. The participation in the initiative has spanned different projects in the organization, and has included projects concerned with strategic planning, architecture and platform development, production system development, IT support, documentation and changes to organization, processes, roles and responsibilities. The researcher has acted in a capacity of active participant, so the majority of the related research activities constitute action research. The research at Grundfos covers a period of 2.5 years, and the setup will be detailed in section 4.7.

Experiments within Danish and European companies

In addition to the research carried out at Grundfos, this PhD dissertation has also involved smaller projects in other Danish and European companies concerned with the design of industrial processing plants. The duration of these projects has been from a few weeks to several months. The projects have both involved study of the design process to determine the requirements for reference designation systems and practical experiments in use of said systems for coding of processing plants.

Funding applications

As part of the PhD study I have participated in two funding applications for EUDP and the Danish High Technology Foundation. The applications were made in collaboration with an industrial consortium within the industry of concrete sandwich elements, and two institutes at DTU. The latter application was successful, and resulted in a total of 6 PhD positions for the two institutes at DTU.

Development of standards for information exchange in the construction sector

In addition to the activities directly related to the research project I have been a member of one of the working groups responsible for developing the new coding rules and principles for referencing building elements and spaces within the Cuneco Classification System (CCS). CCS is a system for classification of information in the Danish building sector and it includes among other a reference designation system for structuring and referencing building elements and spaces. CCS is developed by the industry association *bips*, which is responsible for developing standards, working methods, tools and sector and industry standards for the construction sector as part of a Danish government imitative for increasing the use of information and communications technology in the Danish construction sector, known as *Digital Construction* (DK: Det digitale byggeri).

4.7 Primary research setup

The majority of this PhD study has been carried out within a time period of 2.5 years within Grundfos A/S, where the PhD student has been embedded in a part of the organization called the Technology Centre. The research project has been a key element in a new platform initiative within the design of production systems in Grundfos, and to a very large extent free access has been given to all relevant stakeholders and data both in the domestic and foreign parts of the Grundfos organization.

Grundfos

Grundfos is the world's largest manufacturer of pumps and pump systems, with an annual production of 16 million pumps and revenue of 22 bill. DKK (2012). The company was founded in 1945 and is headquartered in Denmark where most development activities take place. Production is spread out across the world with 14 production companies in Europe, North America and Asia. Today the company has more than 18.000 employees spread out over 56 different countries.

Grundfos Technology Centre (TC)

The Grundfos Technology Center (TC) was established in 1990 as part of the Danish production company. Today TC consists of three organizations: the main office in Denmark, and two smaller satellites in the production companies in China and Hungary, with approx. 170, 35 and 5 employees respectively.

The Technology Center has two main tasks: 1) supply of equipment to the production companies in Grundfos, 2) and development of new production technology. Some service for production systems and prototyping for product development is also carried out, but it is not the main focus.

The Technology Center occupies a special position in Grundfos, in that it is not the guaranteed supplier of equipment for Grundfos production companies. With the exception of key technologies and equipment for certain production processes the production companies are free to use external equipment suppliers. The Technology Center must therefore compete with external suppliers, but does not have the opportunity to sell equipment outside of Grundfos. The volume of production systems developed or procured by the Technology Center is therefore quite low compared to external suppliers, since their market is entirely internal to the company.

As it is, there are several challenges faced by a development organization such as TC, which operates internally in a company. The organization is small relative to the number and variety of production systems needed by the company, and there is a limited 'customer base' relative to external suppliers who sell to several companies. The Technology Center therefore decided to initiate a strategic project, intended to introduce platforms in the design of production systems. Prior to this project platform based design has not been used to a great extent. The main experiences in platform based design at TC have been within production systems for handling, testing and production control. Design reuse also happens ad hoc through the choices of development engineers and by application of internal standards and lists of preferred equipment brands and models, but it is not the result of a common platform strategy.

The role of the PhD student

I have been embedded in Grundfos A/S for a period of 2 years and 5 months, during which approximately 70% of my time has been spent at the company, allowing for participation in platform related projects and observation of the day to day activities of the organization. I have been placed in the Technology Centre (TC) and have primarily functioned as a member of the project team for the strategic platform initiative run by TC concerning the introduction of platform based design for production systems in Grundfos. Throughout the research period I have acted in a capacity as one of three main responsible people for the execution of the strategic project for production system platforms, with the other two members being employees of the Technology Center. In addition to participating in projects for development of production systems and production system architecture and platforms, I have had an active role in shaping the strategic project. The activities of the strategic project can be seen in Figure 2:

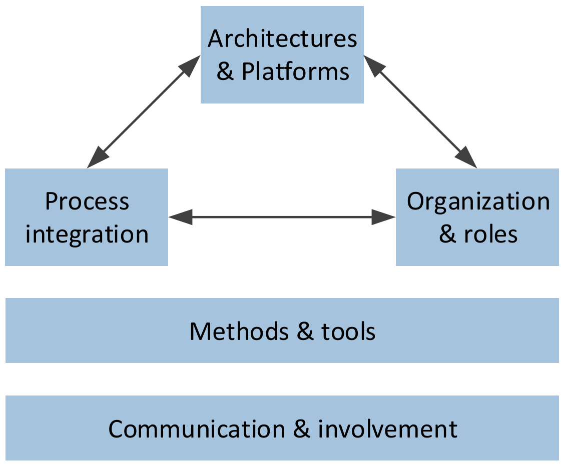


Figure 2 - Activity fields of the strategic platform initiative at the Grundfos Technology Center.

The activities involved in the initiative covers:

- Overall activities related to the entire initiative e.g. formulation of activities and goals, as well as planning and management of the initiative.
- Process integration: Activities related to integration of architecture-centric and platform based working methods in the company's work processes.
- Organization & roles: Activities related to the required changes in organization and changes/establishment of new organizational roles and responsibilities
- Architectures & platforms: Activities related to the development of architectures and/or platforms e.g. as standalone development projects or as a part of equipment procurement/development projects.
- Methods & tools: Activities related to the development and testing of methods and tools for use in all other activities
- Communication & involvement: Activities related to communication of information to the organization and involvement of stakeholders in the company.

As part of the research at Grundfos I have been involved in every one of the activity areas of the initiative, and have had a very active role in shaping the initiative and its activities. This covers among other:

- Participation in development projects for production systems and their architectures & platforms
- Formulating a strategy and execution plan for the strategic platform project
- Communicating the strategic platform project as well as architecture and platform theory to the organization
- Formulating a strategy for development of production system architectures and platforms
- Developing processes for development of production system architectures and platforms
- Developing modeling and documentation tools for production system architectures and platforms
- Defining the necessary organization and roles for architecture & platform based design of production systems
- Participating in the formulation, setup and operation of cross-organizational networks for strategic production technologies. The primary tasks of the networks were knowledge management, technology management, capacity planning and architecture & platform development for production systems

The research in relation to the different activity fields have been conducted as continuous research cycles sometimes isolated to individual activity areas, and at other times overlapping activity areas. At times one cycle of data gathering and analysis in one activity area has feed into planning and implementation of actions in another area, e.g. in the cases where testing of modeling tools have initiated investigation in the organization or processes of the company to assess the usability of the models or prerequisites for their application.

5 Theoretical basis

The intention of this section is to provide an overview of the theoretical foundation which supports the research and to explain how the theories contribute to this research. The theories span both product design and software design, but all have their relevance for research in production system architecture. The following theories provide the foundation of this research.

- Theory of Technical Systems
- Theory of Domains
- Multiple Structures
- Genetic Design Model
- The object oriented paradigm

5.1 Theory of Technical Systems

The Theory of technical Systems (ToTS) represents one of the major contributions to design research and is considered to be fundamental to design science by a large section of the design research community. Several iterations have refined the theory from the early works of Vladimir Hubka (Hubka, 1967) into the present state which provides designers with a theory-based description of technical systems.

ToTS provides a model for describing technical systems. A system in itself can be seen as a concept used for describing sets of related elements. A system is said to consist of a set of related elements and is delimited by a system boundary, which separates the elements from their surroundings. Relations exist between elements of the system and to the elements outside the system. Relations between elements of the systems are what give the system structure. The system concept is recursive, meaning that elements of systems may also be elements in and of themselves (Klir and Valach, 1967). (Eder and Hosnedl, 2007a) describe a hierarchy of systems of which the Technical System is one (see Figure 3). A technical System is a system composed of man-made artefacts that deliver one or more of the effects involved in an operation in a transformation process. The Theory of Technical Systems provides a framework for describing Technical Systems, by viewing Technical Systems as elements of a Transformation System in which a state change for operands takes place through a transformation process.

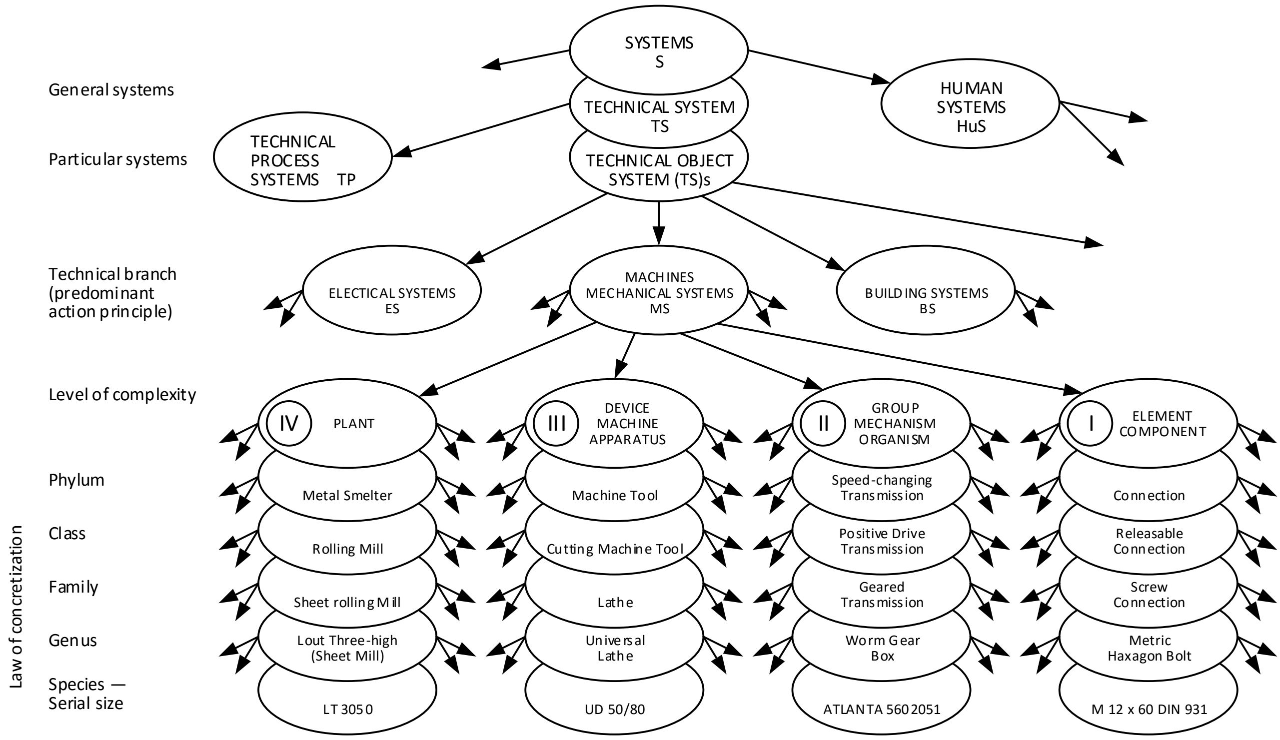


Figure 3 - Hierarchy of technical systems (TS-hierarchy). Figure redrawn from (Eder and Hosnedl, 2007a, p.340)

(Eder and Hosnedl, 2007b) defines a transformation system (see Figure 4) as a system in which:

> *"An operand (materials, energy, information, and/or living things – M, E, I, L) in state Od1is transformed into state Od2, using the active and reactive effects (in the form of materials, energy and/or information – M, E, I) exerted continuously, intermittently, or instantaneously by the operators (human systems, technical systems, active and reactive environment, information systems, and management systems, as outputs from their internal and cross-boundary processes), by applying a suitable technology Tg (which mediates the exchange of M, E, I between effects and operand), whereby assisting inputs are needed, and secondary inputs and outputs can occur for the operand and for the operators."* (Eder and Hosnedl, 2007b, p.1)

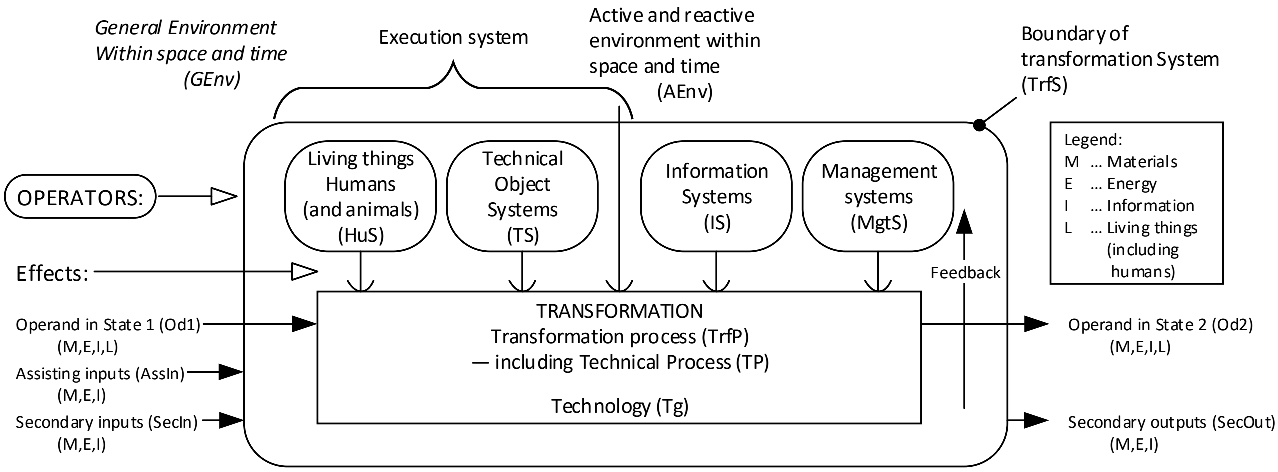


Figure 4 - Model of Transformation System. Figure redrawn from (Eder and Hosnedl, 2007b, p.2)

A Technical Processes (see Figure 5), such as those occurring in production systems, are a class of transformation processes wherein the technical system plays a key role in delivering the necessary effects to carry out the operations of the transformation process. In a Technical Process System the transformation of the operands is primarily obtained by the effects of the Technical System, and the transformation is influenced directly by the Human System and Active & Reactive Environment, and indirectly influenced by the Information System and Management System (Eder and Hosnedl, 2007b).

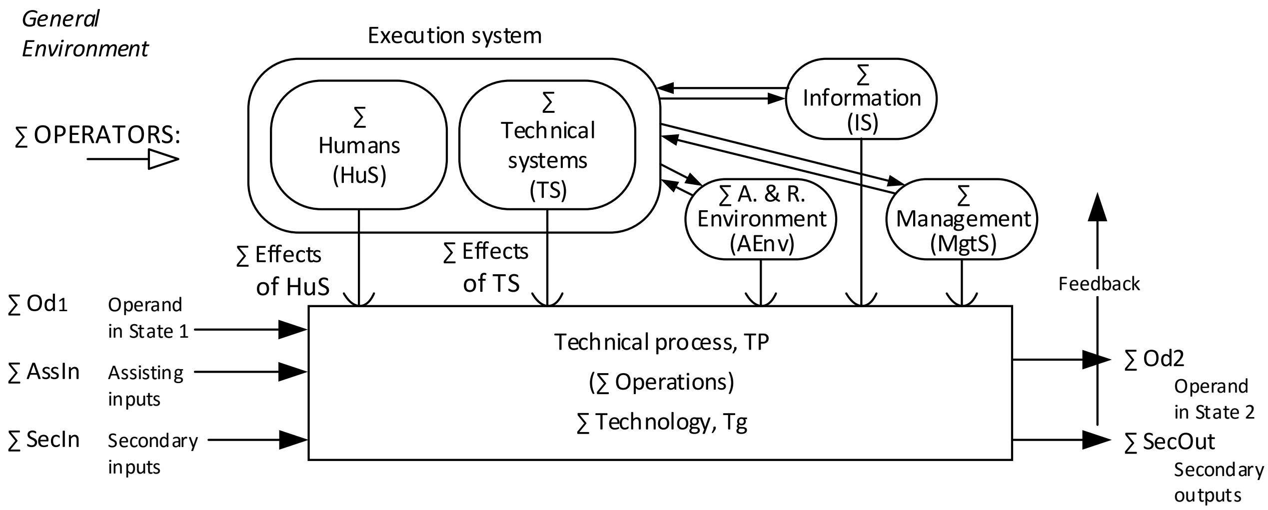


Figure 5 - General model of technical process (system). Figure redrawn from (Eder and Hosnedl, 2007b, p.6)

How the "Theory of Technical Systems" contributes to this research

The Theory of Technical Systems provides the core foundation of this research, as it provides a conceptual framework for analyzing and designing a production system and production system architecture, and a description of what a production system is i.e. a Technical Process System. When describing a product or production system through the use of the Theory of Technical Systems, the perception of what constitutes the subject of design is frequently limited to the Technical System. In this research the definition of production systems (the subject of design), is not limited to the Technical System. Instead the production system as a subject of design is viewed as the entire Technical Process System. Consequently when talking about architectures or platforms within the context of this research, they are the architectures and platforms for transformation systems, and the modeling of production system architecture can include modeling both constitutive and behavioral phenomena related to a transformation system.

5.2 Theory of Domains

The Theory of Domains (Andreasen, 1980) is an extension of the Theory of Technical Systems. Since the Theory was first presented in 1980, it has gone through several iterations. The state of the theory used in this research is based on the current formulation of the theory as advocated by the Technical University of Denmark.

The Theory of Domains provides an interpretation of a Transformation System's constituent elements and their relations, which allows for treating a Technical System in a process of design. The central claim of the Theory of Domains is that the Transformation System may be seen as three different views upon a product, namely an Activity view, Organ view and Part view (see Figure 6).

The three different views represent three different domains, in which the constituent elements of a transformation system exist (Hansen and Andreasen, 2002).

- *Activity domain:* Activities are another name for the transformation processes. It is within the activity domain that the transformation of operands occurs. Activities have also been called transformations in different publications, to be more in line with Theory of Technical Systems.
- *Organ domain:* The organ domain is where Organs are found. Organs are the artefacts that create the effects which enable the operations of the transformation process. Organs are also dubbed "function carriers" or "functional elements" and consist of interacting material elements or material areas.
- *Part domain:* The Part domain (sometimes also called the constructional domain) is where the physical realization of Organs is found. Parts are the artefacts of a product that realize Organs. The specification of material, form, dimension, surface quality and tolerance for the parts and the relation between parts is what creates the organs and their functionality.

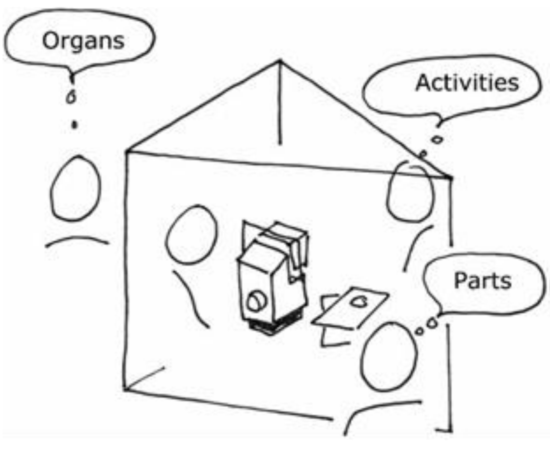


Figure 6 - Domain Theory's three views upon a product and its use activity. Figure from (Andreasen, 2007, p.8)

According to the Theory of Domains the systems existing within the three domains i.e. the Activity System, Organ System and Part System, are considered distinct systems, and not simply abstraction of the same system. (Andreasen, 1980; Buur, 1990; Olesen, 1992) contribute to the notion of the synthesis of products as a process of detailing and concretization of systems within each of the three domains (see Figure 7). Detailing implies that more elements of the artefact are determined, and making the artefact more concrete means that more attributes of the artefact are determined.

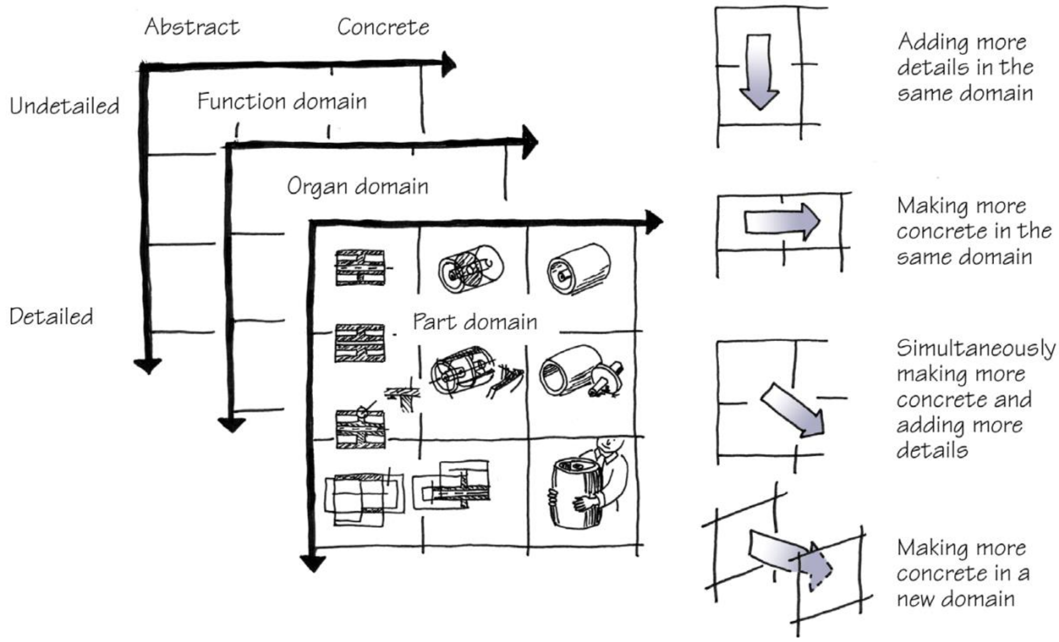


Figure 7 - Product synthesis in the three domains. Figure from (Harlou, 2006, p.80)

How "Theory of Domains" contributes to this research

While the conceptual framework described by Theory of Domains is primarily limited to mechanical products, and seems to only encompass the Technical System and the Technical Process, the basic concepts are sound. A critique of the Theory of Domains would be that it does not offer the distinction between Operators of the Transformation systems which the Theory of Technical Systems provides. In applying the Theory of Domains the domain concept will be applied to the model of a Technical Process System from Theory of Technical Systems. Also, in line with the Theory of Technical systems, I would note that the Artefacts of the Technical System i.e. Organs and Parts, are not limited to physical artifacts, but can be hardware, firmware and software (Eder and Hosnedl, 2007b; Buur, 1990).

The Theory of Domains is applied to the Theory of Technical Systems, by introducing a domain aspect in the description of operators of transformation systems, and by acknowledging the need for describing the operators of a transformation system from different views. The main contributions to be found in the Theory of Domains for this research are therefore:

- Describing a transformation system requires viewing the system from different views.
- Transformation systems are constituted by systems existing in different domains.
- Operators of a transformation system can be constituted by multiple systems or system elements existing in different domains.

5.3 Multiple Structures

The theory of multiple structures deals with the perception of structures in a product. Throughout the life cycle of a product or production system structural information is generated and used by a multitude of stakeholders and IT systems. Being able to describe and analyze constituent elements and their relations is one of the key prerequisites for treating products or production systems in the processes of the company. When design information is generated, processed and shared the underlying structure becomes an expression of how the product or production system is viewed within a company. Very often there is a dominant view of the product or production system structure within a company, typically the physical BOM structure intended for production. But looking at a company where structural information is treated in multiple stakeholder domains, tools, process, etc. it is clear that a single view of the product structure is insufficient to capture and describe the complex nature of a product or production system. It can be stated that *"the structure of a product is the way in which its elements are interrelated in a system model, based on the actual viewpoint"* (Andreasen, Hansen and Mortensen, 1995). This means that structure is dependent on the viewpoint of the observer, and there will be multiple structures within a product or production system. For products (Andreasen, Hansen and Mortensen, 1995) has identified four basic classes of views (see Figure 8)

Synthesis oriented or generic structure views: Synthesis oriented views reflect a chosen process of design synthesis. In (Andreasen, Hansen and Mortensen, 1995) the views in this class are defined according to the Theory of Domains (Andreasen, 1980), representing the domains of the transformation system.

Functional views: Functional views on the product are based on the tasks of the product and are described according to technical disciplines by use of specific domain language e.g. for control, thermodynamics etc.

Product assortment views: Assortment views for products involves structuring based on market related variety and internal commonality. Views of this class reflect the similarities and differences between products.

Product life views: Product life views relate to the "meeting" between the product and the life phases. The views reflect the fit between the product and the other systems it encounters in the life cycle.

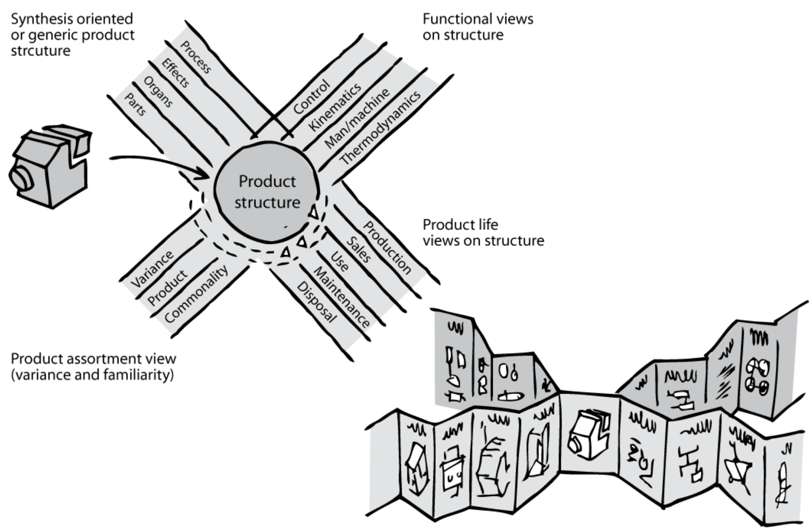


Figure 8 - The totality of product structure views (Andreasen, Hansen and Mortensen, 1996). Figure from (Kvist, 2010, p.43).

How "Multiple Structures" contributes to this research

As architecture-centric and/or platform based design becomes more established within a company, it shapes every part of the company's organization and operation, and therefore requires handling of the architecture or platform by multiple stakeholders and systems. This tendency is magnified by ever more popular design and work methodologies such as Concurrent Engineering, Lean Product development and Systems Engineering to mention a few, which seek to integrate all relevant organizational functions in the development of products or production systems and carry out their development in parallel. It follows that architectures and platforms are not only designed but also applied and maintained simultaneously by multiple systems and stakeholders in the company, which raises the demands for collaboration and exchange between stakeholders and their tools. Multiple Structures introduces the idea that there are multiple superimposed structures for the same production system, which must be documented and communicated between stakeholders and systems, and that these structures are dependent on the viewpoint. The following observations can be made:

- There exists multiple superimposed structures for the same transformation system, or in other words the constitutive systems of a Transformation System can be multi-structural.
- The structures are dependent on the viewpoint of the person or system doing the structuring
- Views can be classified
- There are four classes of views for product structuring suggested by (Andreasen, Hansen and Mortensen, 1996), which could possibly be applied for production systems

The implications for this research are that any modeled structures depend on the viewpoint. Because there are many different viewpoints on the production system, there is a need to express different element relations and system boundaries (different structures) i.e. there is a need for modeling and communicating different structures for a production system.

5.4 Genetic Design model System

The Genetic Design Model System (GDMS) (Mortensen, 1999) presents a system for describing models related to the synthesis of designs with consideration for the product life cycle. The chromosome model of GDMS incorporates elements of Theory of technical Systems, Theory of Domains and Theory of dispositions (Olesen, 1992), which describes the dispositional relations between a products design and the life cycle phases. GDMS presents a distinction between models describing the design itself and models describing the meeting between the design and the life phases.

Constitutive models: Models which describe a design independent of the meeting between the design and the life phases are called "constitutive model". Models of this kind answer the question "what is it?"

Behavioral model: Models which describe the meeting between the design and life cycle phases are called "behavioral models. Models of this kind answer the question "what does it do?"

Constitutive models in GDMS are based on the three viewpoints from Theory of Domains, Organs, Parts and Activities. In the case of the Activity viewpoint, the modeling unit is the technology which is used to create the effect, rather than the Activity itself, since the Activity takes place in the meeting between the design and the life-phase and therefore belong to the behavioral models. Behavioral models in GDMS are divided into “Soll” and “Ist” models (see Figure 9). Soll behavior has to do with what a design should do, while Ist behavior describes the actual resulting behavior.

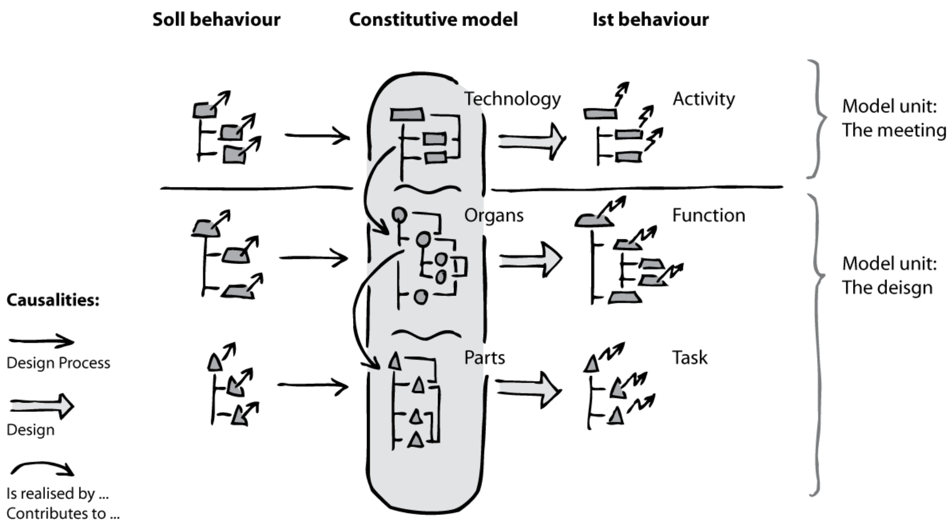


Figure 9 - Constitutive and behavioral aspects of the chromosome model (Mortensen, 1999). Figure from (Kvist, 2010, p.45).

There is a clear distinction between the attributes of design elements of models in GDMS, depending on whether or not the attributes relate to the constitution of the design or the behavior of the design. Attributes that relate to the constitution of the design are called “characteristics” and attributes that relate to the behavior of designs are called properties. Properties are furthermore separated into Inherent properties and relational properties. The three concepts can be described as:

Characteristics: Attributes of design elements in constitutive models are called “Characteristics”. They are the answer to “what is it?”, and are the attributes of design elements which can be directly determined by designers. (Tjalve, 2003) defines four characteristics for mechanical design elements: Form, material, Dimension and Surface quality.

Inherent properties: Inherent properties are those properties which are intrinsic to the design such as weight and strength. The inherent properties are determined by the characteristics of the design and the environment.

Relational properties: Relational properties describe the behavior of the design in the meeting between the design and the life phases i.e. they are the properties of the design which arise when the design goes through its life cycle such as cost and quality.

How “GDMS” contributes to this research

GDMS affects this research by emphasizing the need for distinguishing between constitutive and behavioral aspects of a design. When modeling production system architecture, it could be necessary to model both constitutive phenomena and behavioral phenomena.

5.5 The object oriented paradigm

The object-oriented paradigm is a model for abstraction that allows for conceptualizing real world artefacts in terms of discrete objects that have an associated state, behavior and identity. The paradigm has emerged from many different fields of computer science, and today it constitutes the foundation of object-oriented software engineering which deals with the analysis, design and implementation of software and computer systems (Booch et al., 2007, p.39). As time has passed the object-oriented paradigm has also found more widespread use outside the field of computer science as a general model for analysis and design of complex systems. Methods, tools and activities encountered in design of such systems are said to be object-oriented if they are based on the same conceptual framework that underpins the object-oriented paradigm. This framework is known as the Object Model and it consists of four main concepts:

Abstraction: *"An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer"* (Booch et al., 2007, p.44).

Encapsulation: *"Encapsulation is the process of compartmentalizing the elements of an abstraction that constitute its structure and behavior"* (Booch et al., 2007, p.52), and *"encapsulation hides the secrets of the implementation of an object"* (Booch et al., 2007, p.51). *"Encapsulation is most often achieved through information hiding (not just data hiding), which is the process of hiding all the secrets of an object that do not contribute to its essential characteristics; typically the structure of an object is hidden, as well as the implementation of its methods"* (Booch et al., 2007, p.51).

Modularity: "Modularity is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules" (Booch et al., 2007, p.56), and it is said that modularity provides "a way to cluster logically related abstractions" (Booch et al., 2007, p.58). In other words modularity packages abstractions into discrete units that can be used to decompose the system-of-interest.

Hierarchy: "Hierarchy is a ranking or ordering of abstractions" (Booch et al., 2007, p.58). Hierarchies define a relationship between abstractions in the system-of-interest. The two most important hierarchies are the kind-of (is-a) and part-of hierarchies

One of the key applications of the object-model within object-oriented software engineering is in the object-oriented analysis and design where object-oriented modeling provides the input for object-oriented programming. An object-oriented model provides the input for object-oriented programming, and must conform to the three core characteristics associated with object-oriented programming. (Booch et al., 2007, p.41) states that the basic characteristics of object-oriented programming are that it *"(1) uses objects […]; (2) each object is an instance of some class; and (3) classes are related to one another via inheritance relationships".* The concepts of object, class and inheritance are defined in literature as follows:

Object: *"An abstraction of something in the problem domain, reflecting the capabilities of a system to keep information about it, interact with it, or both; an encapsulation of attribute values and their exclusive Services"* (Coad and Yourdon, 1991, p.53). *"An object is an entity that has state, behavior and identity. The structure and behavior of similar objects are defined in their common class. The terms instance and object are interchangeable"* (Booch et al., 2007, p.78). *"A single object is simply an instance of a class"* (Booch et al., 2007, p.93).

Class: *"A class is a set of objects that share a common structure, behavior, and common semantics"* (Booch et al., 2007, p.93). In common usage *"a group, set, or kind sharing common attributes"* (Merriam-Webster, n.d.)

Inheritance: Inheritance is the mechanism by which classes may inherit structure and behavior from other classes in an inheritance relationship, also called a generalization-specialization relationship (Booch et al., 2007, pp.58–59). The generalization-specialization relationship is also known as a "kind-of" or "is-a" relationship, where one class is a subtype of another class e.g. a Dog is a "kind-of" Animal.

How the object oriented paradigm contributes to this research

The object-oriented paradigm can be combined with many of the artefact theories applied in the field of engineering design, and indeed share concepts with many of the theories. When the object-oriented paradigm is combined with artefact theories such as Theory of Technical Systems and Theory of Domains it provides a useful means of transitioning from phenomenon models to information models that can aid in the description of production system architecture. One of the key contributions to this research from the object-oriented paradigm lies in the conceptual framework it provides when creating models of the production system, indeed the model kinds that are applied or developed in the description of production system architecture are often object-oriented models.

In relation to engineering design in general I would perceive something as being object-oriented if it uses abstraction to define objects based on their defining characteristics; uses encapsulation to hide/disregard the non-essential information of the object; allows for clustering of abstractions based on their logical relations; and can address (describe, handle, etc.) the hierarchies of abstractions inherent in the real world object-of-interest. For the purpose of this research a model can be said to be object oriented if it conforms to the three characteristics associated with object-oriented programming.

6 Conclusion on setting the stage

Part 1 of this dissertation has set the stage for the research in terms of the research motivation and underlying theoretical basis. The reader should now know what the expected goals and scientific contributions of the research are, and what research activities have been carried out.

From Part 1 it can be concluded that there is a need to address a great number of challenges for production companies, particularly in high cost countries such as Denmark. Experience from the product design research field has demonstrated that architecture & platform based product design can contribute to more efficient and profitable development of products. It is believed that the same could be achieved in the area of production system design if an architecture-centric design approach is adopted, to strengthen existing design reuse practices. It has also been established that in order for this to become reality, there is a need for greater understanding of the architecture phenomenon in relation to production system design; and a need for new or modified supporting modeling and information handling tools.

Part 2 of the dissertation will investigate the architecture phenomenon within the field of production system design and will expand upon the understanding of the concept.

Part 2: A Contribution to a theory of production system architecture

Part 2 investigates the architecture phenomenon for production system design and contributes to the existing vocabulary within architecture & platform related design theory. A framework is introduced that describes architecture for production systems as a layered phenomenon. The framework consists of two main layers describing the production system design and the production system application; and the connection between the design and application. The architecture design layer is concerned with the constitutive and functional behavioral aspects of the production system i.e. what the system is and does; the architecture application layer is concerned with what the production system is used for and how it affects and is in turn affected by the combined doctrines, strategies, tactics and operations of a company. The framework serves two purposes. First and foremost it provides a basic framework around which to investigate the architecture phenomenon and the requirements for the desired design and information handling tools for architecture-centric design. Secondly it provides a conceptual frame of reference for companies engaging in or transitioning to architecture-centric design of production systems. This can be useful when discussing concepts, plans, goals, priorities, processes, tools, organization etc. in the context of strategic, tactical, and operational decision making. Part 2 furthermore relates the terminology and concepts of production system architecture to established concepts within product design and systems engineering.

7 A note on the term production system

The simplest perception of a production system (or manufacturing system) is that of a collection of various machines, e.g. equipment for material processing, assembly, handling testing and inspection, etc. as well as the systems controlling and managing the equipment. Broader perceptions may see it as a set of manufacturing practices such as the American system of manufacturing. In the context of this research a production system is defined as a transformation system in accordance with the Theory of Technical systems, and consequently when referring to a production system this includes both the operators of the system and the production process, seen as an element of the system.

Allow me also to make a note on the use of the terms “production” and manufacturing in this research project. These two terms are unfortunately used in an inconsistent manner in both the academic and industrial community. Sometimes they are used interchangeably, and at other times they are given a very specific definition in relation to each other. The two terms are separated in many, often contradictory, manners according to:

- The types of processes covered
- The extent of the process chain covered
- The means used in the systems
- The system levels
- The method of organization

As an example manufacturing is sometimes seen as a kind of production, and at other times it is used to describe a method of organizing production e.g. production systems organized in a company´s global manufacturing system. The inconsistent use of the terms can be confusing, so for the purpose of consistency I will primarily use the term production throughout this research, including in places where others might use the term manufacturing. In this research there is no need or benefit to make any distinction, and as such the two terms can be considered to be interchangeable in regards to both my own contributions and in the cases where other sources of theories, models etc. are involved. In the case where manufacturing might be used, it is done simply for the sake of not changing commonly used names for concepts, phrases, etc.

8 Types of architecture

Any vocabulary for architecture within production system design should be related to the existing vocabulary within product design and systems engineering, in order to strengthen reuse and correlation of concepts across these related design fields. However, although the concept of architecture has been widely used within the field of product design and development for several years, there is no single commonly agreed upon definition for the concept. When applying the architecture concept to the field of production system design, it therefore begs the questions: What is the current range of definitions and how do they apply to production system design? What are the shortcomings of existing definitions? What are the similarities and differences between the architecture and platform concept in product design, systems engineering and production system design? Attempting to develop supporting tools for architecture-centric design without a proper investigation of these questions will be difficult, and it is fundamental in the formulation of requirements for the modeling and information handling tools developed as part of this research. Simply put, if the modeled concepts and phenomena are not defined, then no such tools can be developed and tested. The purpose of this section is therefore to establish an understanding of the concept of architecture for production systems and to formulate a working vocabulary. The following subsections will present some of the most common definitions and uses of the architecture concept within related design fields, and will demonstrate the need for expansion and adjustment of the architecture concept to fit the needs for a vocabulary within architecture-centric design of production systems.

To understand what production system architecture is it is important to know that there are different perceptions of the basic architecture concept and its qualifier ‘production system’. In Section 7, it was explained that the production system as far as this research is concerned, can be defined as a Transformation System in accordance with the Theory of Technical Systems, and that the concept of a production system in this research is not limited to the Technical System. The concepts of architecture on the other hand are not so easy to define. The use of the terms architecture in the context of design is wide and varied; and they cover many different design disciplines. Indeed the list of qualifiers associated with architectures and platforms is very long e.g.:

- Product (architecture)
- Systems (architecture)
- Software (architecture)
- Hardware (architecture)
- Enterprise (architecture)

- Process (architecture)
- Information (architecture)

The most relevant qualifiers for the definition and understanding of the concept are ‘Product Architecture’ and ‘Systems Architecture’ originating in the fields of Product Design and Systems Engineering respectively. By combining concepts from these two fields and viewing the Production System in the context of Theory of Technical Systems, it is possible to set up a conceptual framework for the Production System Architecture phenomenon that will both provide a theoretical understanding of the phenomenon, and a guiding architecture framework for companies engaged in/or transitioning to architecture-centric and/or platform based design of production systems. The reason that systems architecture is investigated rather than merely production system architecture, is that there is a broader body of work available for the more general systems architecture concept than for the concept of production system architecture, and because most of the literature available on production system architecture is focused on the underlying production philosophy, production tactics, or performance capability of the production system, etc. rather than the fundamental nature of what constitutes a production system architecture. As such it is more valuable to look into the definitions of systems architecture in general.

The following two sections will present the most common understanding of the architecture phenomenon within the two fields, and use the available concepts as a basis for the architecture definition of this research.

9 Product architecture in literature

Product architecture has been the subject of a rapidly expanding body of research within the product design community, and perhaps as a natural consequence, there is a great difference in perception of the concept. Two issues must be considered to gain a proper understanding 1) what is the fundamental nature of product architecture and 2) what phenomena does the concept cover. Few authors make it clear what the fundamental nature of product architecture is. From the different definitions and descriptions of the architecture phenomenon present in literature, it is clear that a product architecture is seen either as an inherent aspect of a product or a means of describing a products design. As an example (Meyer and Lehnerd, 1997) states that all products have architecture, while (Harlou, 2006) defines product architecture as a structural description of a product. As far as this author is concerned, product architecture can be considered to be something inherent to a product, whereas the description of product architecture is something different entirely. What follows are some of the main observations from literature regarding the concept of product architecture.

Architecture as principle partitative structure

In terms of what the product architecture concept covers, the range of definitions both within the scientific community and in industry is almost as wide as the research subjects associated with the term. Still, it is evident that a great many researchers adhere to a simple view of product architecture, in which product architecture is seen as being near synonymous with the principle partitative structure of hardware and software of a product, product family or product assortment, i.e. the partition and configuration of hardware and software parts. In these cases the architecture is also frequently given a time aspect, and is seen to express the changeability of the principal partitative structure. Several definitions build from this starting point to include other phenomena to provide more detail or cover more design phenomena. (Meyer and Lehnerd, 1997) represents one such understanding of the term architecture, stating that all products have an architecture, and that *"the combination of subsystems and interfaces defines the architecture of any single product"*. The architecture is thereby seen as equal to the principle partitative structure of the product, and includes only the elements implementing functionality and their interfaces. They further define a product platform architecture as an architecture which is designed and used as the basis for creating several more derivative products. This perception of architecture as the principle partitative structure is often used when either talking about the architecture in a narrow focus of the design e.g. exclusively in the context of structural variety across products; when the product architecture itself is not the main subject of research; or when the term is used by stakeholders outside of the design field. We can ignore this limited use of the term, since it does not offer any significant difference compared to the concept of the partitative product structure except to include the interfaces between parts. Instead I will focus on broader understandings of the concept.

Architecture as functional implementation scheme

One of the most commonly agreed upon definitions of product architecture is proposed by (Ulrich, 1995) who defines product architecture as: *"(1) the arrangement of functional elements; (2) the mapping from functional elements to physical components; (3) the specification of the interfaces among interacting physical components."* In this definition (Ulrich, 1995) offers an additional aspect to the partitative structure in the inclusion of functional elements. The architecture in Ulrich's definition is thus not only a structure of hardware, but also an expression of the functional structure of a product, and the scheme by which the functionality is implemented through parts. The phenomenon is later detailed in (Ulrich and Eppinger, 2004).

There exist variations and extensions of Ulrich's definition. The common denominator for these variations is the basic principle exemplified by Ulrich's definition i.e. that product architecture is a structural description of the functional elements; the structure of the design elements implementing the functions; and the mapping scheme describing the allocation of functions to implementing design elements. In this way the product architecture is seen as a description of the transformation from function to the decomposition and configuration of implementing parts. Variations and extensions of the definition expressed by (Ulrich, 1995) are seen to cover among other the following phenomena:

- Organs
- Software & firmware
- Multiple products

Architecture includes Organs

The Product Architecture Group at the Department of Mechanical Engineering at the Technical University of Denmark has worked extensively with research in architecture and platform based design of products for many years. The prevailing definition of product architecture within the research group is exemplified by (Harlou, 2006) which defines product architecture as:

> *"An architecture is a structural description of a product assortment, a product family or a product. The architecture is constituted by standard designs and/or design units. The architecture includes interfaces among units and interfaces with the surroundings."* (Harlou, 2006)

(Mortensen and Andreasen, 1996) defines design units as:

> *"A design unit is a function, organ, part or an encapsulation of a group of these. The design units together constitute a product."* (Mortensen and Andreasen, 1996)

This definition has its base in both (Ulrich, 1995) and the Theory of Domains. The Theory of Domains offers an analogue definition to (Ulrich, 1995), where organs are included as the intermediate elements connecting functions (or activities/transformations as defined by Theory of Domains) and parts. This is a consequence of the Theory of Domains expanding upon the definition of a product viewed as a technical system, by including functional carriers i.e. Organs, as an intermediary between transformations and components. This definition recognizes the multifunctional nature of individual parts, and it demonstrates that functionality mostly exists through the interaction of multiple elements and not single parts. The inclusion of Organs as elements of the architecture further expands upon the mapping between the functional elements and parts as suggested by Ulrich, and it acknowledges that parts can be instrumental in implementing multiple functions.

Architecture as hardware, software and firmware

One way in which the view of product architecture differs from that of (Ulrich, 1995) in the context of this research project is in the inclusion of software and firmware in the definition of the elements of a product architecture. (Buur, 1990) points to the fact that function carriers (organs) can also be comprised of software elements, but that software can not in itself constitute organs. Only in conjunction with hardware can software implement functionality and carry out an activity (i.e. execute a process). This suggests that software elements are parts but exist in a class of their own separated from that of physical parts. (Eder and Hosnedl, 2007b) also notes that within the Theory of Technical Systems the Artefacts of the Technical System i.e. Organs and Parts, are not limited to physical artefacts, but can be hardware, software and firmware (Eder and Hosnedl, 2007b).

The focus of this dissertation is not on the software elements of production system architecture or platforms, except to say that software and firmware is considered an element of production systems alongside hardware, though in a class of its own. There are thus two main classes of parts physical and non-physical parts (of which software is one). Whether physical and non-physical parts exist in separate domains as seen by the Theory of Domains is not determined, and is not considered important in the context of this research project, since software is not the focus of the research.

Architecture for multiple products

Many sources recognize architecture as being recursive in nature or applying to more than individual product designs. Representative examples are presented here.

(Harlou, 2006) frames the architecture phenomenon as a recursive phenomenon that allows for description of architecture for associated product designs (defined as product families or product assortments). According to Harlou you can therefore talk about both a product architecture, product family architecture and product assortment architecture. The observation by (Harlou, 2006) of architecture as a recursive phenomenon differs from other definitions in the focus on design units, and the aspects of design sharing and reuse across multiple products, product families and product assortments. (Harlou, 2006) describes design units as being either non-standard i.e. specific to a single product variant, or standard i.e. shared between more than one product variant. The theory thereby defines a sharing aspect of the architecture and relates this to the concept of a platform, by describing the architecture as the structural description of a product, product family or product assortment, which enables description of sharing phenomena for design units between products. In this sense the collection of existing standard design units and their interfaces constitute the platform.

(Yu, Gonzalez-zugasti and Otto, 1999) defines a portfolio architecture based on customer demands, and (Meyer and Lehnerd, 1997) defines the architecture for multiple products in the context of platform design by stating that an architecture can constitute a *"product platform architecture"* if it is designed and used as the basis for creating more derivative products. (Martin and Ishii, 2002) also describes how architectures exist for product families by saying that:

> *"A product family can also have an architecture. A family architecture implies that the different products have a common arrangement of elements, common mapping between function and structure, and common interactions among components. A product family architecture only exists if this commonality is present."* (Martin and Ishii, 2002)

(Tseng and Jiao, 1998) relates the phenomenon of an architecture to multiple products through the existence of platforms from multiple products, a Product Family Architecture as they call it:

> *"In essence, a PFA means the underlying architecture of a firm's product platform, within which various product variants can be derived from basic product designs to satisfy a spectrum of customer needs related to various market niches. In other words, a good PFA provides a generic architecture to capture and utilize reusability, within which each new product instantiates and extends so as to anchor future designs to a common product line structure."* (Tseng and Jiao, 1998)

It is clear from these examples that product architecture is not limited to single products, but that the architecture phenomenon for products can also express sharing and commonality between multiple products. In these definitions the architecture phenomenon is therefore not only descriptive of the design of individual products it is also the scheme by which sharing occurs between multiple products.

Conclusions from a review of product architecture literature

From the review of literature for product design it is clear that many different authors either present an individual definition of what constitutes an architecture or adds additional aspects to the understanding of the concept. The product architecture concept is therefore seen as a multi-facetted phenomenon covering one or more of the following aspects:

- Functional structure (functional elements and their relations)
- Organ structure (functional carriers i.e. the intermediate artefacts constituted by parts which implement functionality)
- Part structure including interfaces between parts
- The mapping scheme between functional structure, organ structure and part structure
- Parts can be physical (e.g. components & modules) or non-physical (e.g. software & firmware)
- The architecture is recursive i.e. architectures or elements of architectures can be shared across multiple products thereby defining e.g. product family architectures or product assortment architectures

All of these aspects of product architecture are also seen as true of a production system architecture. In total it can be seen that a product architecture not only expresses the principle partitative structures of products, but also further relations between the constituent elements of products such as functional elements and physical parts implementing the functions. It could be said that product architecture as a phenomenon expresses the structures of products, inter-domain relations (e.g. physical part structure) and intra-domain relations (e.g. mapping relations between function elements and organs, and between organs and parts). Additionally architecture is seen as an expression of the commonality of structure and elements between products.

For further review of literature for product architecture definitions see (Jiao, Simpson and Siddique, 2007) which offers an overview of key contributors within the design research community and demonstrates that there is a greater nuance to the product architecture definition from other sources.

10 System architecture in literature

The concept of 'system architecture' is commonly used both within the scientific community and in the private and public sector, most commonly in relation to activities within the discipline of Systems Engineering, Software Engineering, Manufacturing Systems Engineering and general design of complex systems. Systems engineering is in this case the most relevant for the purpose of investigating the architecture phenomenon for production systems, since production system design fits well into the domain of systems engineering. Systems engineering is characterized by the focus on architecture-centric design for the creation of complex engineered systems. It is described variously as a perspective on system design, a process and a profession. The international professional organization International Council on Systems Engineering (INCOSE) defines Systems Engineering as:

> *"Systems Engineering (SE) 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, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs."*(INCOSE, 2011, p.7)

Systems engineering is distinguished from traditional engineering design in part by the focus on architectural design. In this section representative examples of the use of the systems architecture concept within systems engineering will be explored from the perspective of the scientific community and the public and private sector.

10.1 System architecture in the scientific community

Within the scientific community system architecture is described variously as an inherent aspect of systems or a description of the key characteristics of the systems composition, functionality and behavior, as well as the rules and guidelines governing the design. This difference in perception of the architecture phenomenon represents a basic distinction of what system architecture is. One perception sees system architecture itself as something inherent to a system, while another perception sees system architecture as a description or means of describing certain key system aspects. These two perceptions are not equal, and any definition of production system architecture should make it clear if architecture is something inherent to the system or a means of describing certain system related aspects when conducting design. Making the distinction is not trivial, since there are differences in opinion of what exactly a system's architecture includes, and what should be described when designing a system's architecture. Examples of this difference in perception are given in the following.

System architecture as an inherent aspect

According to (Tripathy and Eppinger, 2007) systems architecture is something inherent to a system and relates to the structures of a system:

> *"The sub-systems within the system and the components within a sub-system are interconnected or dependent on each other and these relationships define the system architecture. Complexity of a system is defined by the complexity of the interconnections and/or the dependencies in the system architecture. Architecture therefore relates to the structure – in terms of components, connections, and constraints - of a product, system, process, or element."* (Tripathy and Eppinger, 2007, pp.5–6)

System architecture as a system description

(Golden, 2013) offers an example of System Architecture as a description of certain system characteristics by suggesting that the architecture of a system (similarly to the one of a building) is a global model of a system consisting of:

- *"a structure*
- *properties (of various elements involved)*
- *relationships (between various elements)*
- *behaviors & dynamics*
- *multiple views of the system (complementary and consistent)."* (Golden, 2013, p.14)

System architecture as both inherent aspect and system description

In contrast to (Tripathy and Eppinger, 2007) and (Golden, 2013) the work by (Crawley et al., 2012) exemplifies the sometimes inconsistent perception of architecture, by describing system architecture both as an inherent aspect of the system and a means of describing aspects of the system design, operation and management. According to (Crawley et al., 2012) *"Systems have multiple architectures and hierarchies of architectures"*, and at the same time *"system architecture is an abstract description of the entities of a system and the relationships between those entities"* and *"…'rules to follow when creating a system' conveys coordination rules, so that different people at different times and places can create systems that are compatible in various ways"*. Architecture in the perception of (Crawley et al., 2012) is thus both something inherent to a system and a description of inherent system characteristics and the means of achieving certain design, operation and management goals. In total (Crawley et al., 2012) treats system architecture variously as:

- A way to understand complex systems
- A way to design complex systems
- A way to design standards and protocols to guide the evolution of ling-lived systems
- A way to manage complex systems

(Crawley et al., 2012) thus offers a good example of how the term system architecture is given different meaning depending on what aspects of the system architecture phenomenon is the subject of research.

System architecture as constitutive and functional behavioral aspects

Regardless of whether system architecture is seen as an inherent aspect of the system or a description of certain system aspects, an overlap with the product architecture concept can often be seen in in terms of what aspects of a system is covered by the architecture phenomenon. This is exemplified by Levis (1999 cited in Crawley et al., 2012, p.5) who asserts that multiple architectures are involved in the architecting process. Namely:

> - *"The functional architecture (a partially ordered list of activities or functions that are needed to accomplish the system's requirements)*
> - *The physical architecture (at minimum a node-arc representation of physical resources and their interconnections)*
> - *The technical architecture (an elaboration of the physical architecture that comprises a minimal set of rules governing the arrangement, interconnections, and interdependence of the elements, such that the system will achieve the requirements)*
> - *The dynamic operational architecture (a description of how the elements operate and interact over time while achieving the goals)"* (Levis, 1999 cited in Crawley et al., 2012, p.5)

In order to distinguish the architecture concept from that of structure these so-called architectures could more appropriately be seen as different structures covering the same key constitutive and functional behavioral aspects of a system most commonly included in definitions of product architecture.

10.2 System architecture in the public and private sector

The term 'system architecture' is broadly used within industry, but particularly within the field of Systems Engineering. It is often encountered within large scale or highly complex engineering designs, for example within engineering design for defense, infrastructure and industrial facilities. However as with product architecture different organizations have different definitions of what constitutes architecture and how to describe it. Some of the most relevant organizations include but are not limited to Standardization organizations, Government Agencies and Professional organizations. Some key examples are given in the following:

Standardization organizations

In addition to offering many standards for specific architecture applications, the ISO, IEC and IEEE standardization organizations, offer a joint designation standard for architecture descriptions within Systems and Software Engineering i.e. the ISO/IEC/IEEE 42010:2011(e). This standard is not itself a specification of how architectures within these areas should be designed, but rather a standard for the content and formulation of descriptions of such architectures. ISO/IEC/IEEE 42010:2011(E) offers the following definition of architecture:

> *" fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution"* (ISO, IEC and IEEE, 2011)

The standard also specifies that:

> *"The architecture of a system constitutes what is essential about that system considered in relation to its environment. There is no single characterization of what is essential or fundamental to a system; that characterization could pertain to any or all of:*
>
> *a) system constituents or elements;*
> *b) how system elements are arranged or interrelated;*
> *c) principles of the system's organization or design; and*
> *d) principles governing the evolution of the system over its life-cycle."* (ISO, IEC and IEEE, 2011)

This definition offers one of the broadest views of what constitutes architecture, and it allows for adaptation of the fundamental architecture concept to many different design applications including the design of products and production systems. Just as importantly it offers a clear distinction between the architecture as an aspect of a system and the architecture as a description of certain key characteristics of a system. The standard does this by distinguishing between an architecture and an architecture description, where it is said that "*architecture descriptions are used to express architectures for systems of interest"* (ISO, IEC and IEEE, 2011). In this definition the architecture is something fundamental to all systems, while an architecture description is something which is used to express said architecture (see Figure 10).

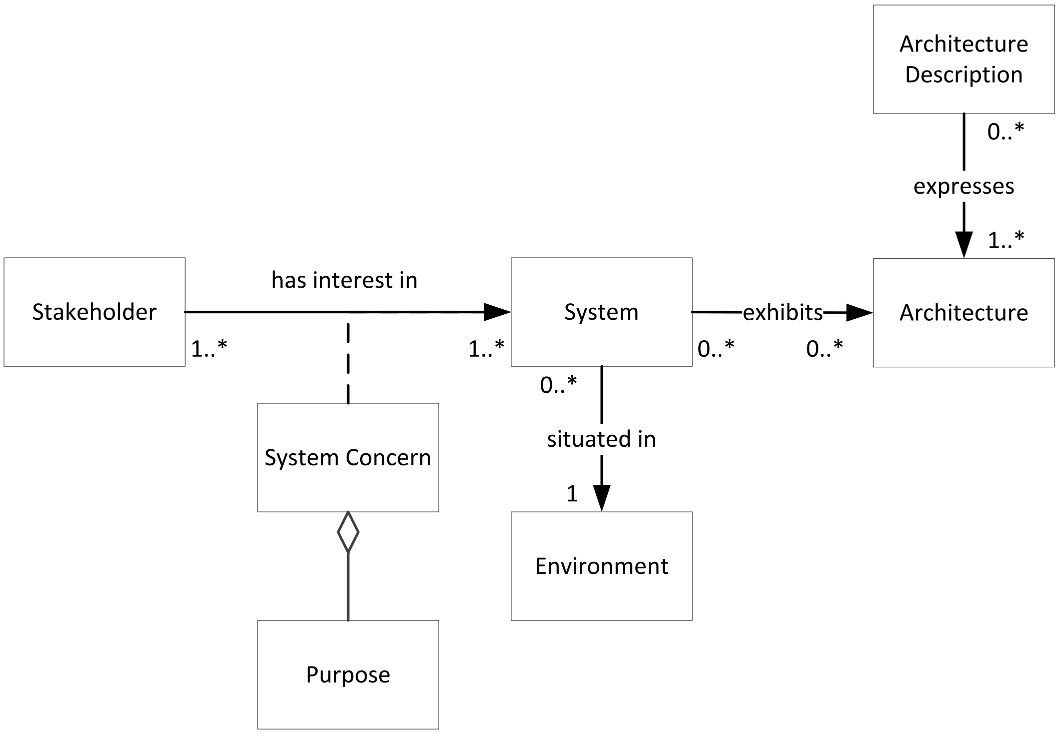


Figure 10 - Context of architecture description. Redrawn from (ISO, IEC and IEEE, 2011, p.3).

It is worth noting that the definition of architecture in (ISO, IEC and IEEE, 2011) also includes principles for the evolution of the system during its life-cycle. The evolution is dependent on the various applications the system is intended for in the company, and this would therefore support that there is an application aspect to the architecture phenomenon.

Professional organizations

Two contributions to the understanding of system architecture from one large, and one small professional organization will be reviewed here.

The International Council on Systems Engineering (INCOSE) is one of the largest professional organizations for systems engineers in the world and its members can be found in both the private industry and the academic community. The organization's views on systems engineering are expressed in the INCOSE Systems Engineering Handbook (INCOSE, 2011), which provides a guide for systems engineers to the systems engineering discipline. The handbook does not directly provide a definition of architecture, but follows the ISO/IEC 15288 standard in the formulation of the Systems Engineering Process. (ISO and IEC, 2008) in turn uses the architecture definition from (ISO and IEC, 2007) which defines system architecture as:

> *"fundamental organization of a system embodied in its components, their relationship to each other, and to the environment, and the principles guiding its design and evolution"* (ISO and IEC, 2007)

This definition has since been altered in ISO/IEC/IEEE 42010:2010(E) which provides a broader definition not limited to a component view of a system (as explained in the previous section), but this has not been fully reflected in the INCOSE SE handbook. In general the INCOSE definition a view where system architecture is still largely component focused and equal to the structure of the physical and non-physical components (hardware, software and firmware) implementing functionality. The SE handbook states that system architecture is characterized by the following

> *"…capture the logical sequencing and interaction of system functions or logical elements."* (INCOSE, 2011, p.98)
>
> *" Identify interfaces and interactions between system elements (including human elements of the system) and with external and enabling systems."* (INCOSE, 2011, p.98)
>
> *"In his book Systems Analysis, Design and Development, Charles Wasson states, 'System, product, or service architectures depict the summation of a system's entities and capabilities at levels of abstraction that support all stages of deployment, operations, and support'."* (INCOSE, 2011, p.99)
>
> *"… System Architecture (defined as the selection of the types of system elements, their characteristics, and their arrangement)…"* (INCOSE, 2011, p.100)

The INCOSE definitions can be seen as being very much in line with the definitions seen in product design. In the INCOSE SE handbook this is evidenced by the presence of two architectures the 'physical architecture' and 'functional architecture'. These architectures can be viewed as analogue to the physical and functional structures of a system.

In addition to adhering to the definition of architecture from ISO/IEC 15288:2007, INCOSE has sought inspiration in definitions from other standards, organizations and agencies. The INCOSE Concept and Terms Working Group (INCOSE, 1998) identifies the following definitions:

> *"The structure of levels and/or branches that partition a system into its constituent parts or components. (DERA – Defense Evaluation and Research Agency)"*
>
> *"The design and interconnection of the main components of a hardware/software system. (DSMC - Defense Systems Management College)"*
>
> *"The organizational structure of a system or component. (IEEE 610.12-1990)"*
>
> *"How functions are grouped together and interact with each other. (NASA MDP92)"*
>
> *"A logical or physical representation of a product which depicts its structure, but, provides few or no implementation details. (IEEE P1220)"*
>
> (INCOSE, 1998, p.19)

These definitions follow the same pattern of definitions seen for the simple definitions for product architecture. They define architecture as a structural description of a system or product encompassing both functional and physical elements, as well as the scheme by which the two are connected. This further emphasizes the key role of the functional elements and parts in the description of architecture.

Unlike INCOSE the not-for-profit OPEN Consortium takes a broader view of architecture. The OPEN Consortium is a much smaller international group of methodologists, academics, CASE tool vendors and developers. Despite its relative small size compared to INCOSE, the consortium none the less provides an expansive process framework which includes an expanded understanding of what design phenomena are covered by the architecture concept. The OPEN Process Framework (OPEN Process Framework Repository Organization, 2009) described by the OPEN Consortium defines architecture as a work product which:

> *"captures the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior of an:*
>
> - *Business Enterprise,*
> - *System,*
> - *Application,*
> - *Application Framework, or*
> - *Reusable Component."*
>
> (OPEN Process Framework Repository Organization, 2009)

Although this view of architecture sees architecture as a description rather than something inherent to systems, the OPEN process Framework, provides an important understanding of the architecture concept. In the view of the OPEN Process Framework architecture is not simply related to the structural aspects of a system as described by INCOSE, but it is a means of capturing inventions, decisions and rationales of design. In other words the architecture not only describes the constitutive and functional behavioral aspects of a system, but is also a description of solutions, decisions and rationales related to the intended application of the system. The architecture phenomenon according to the OPEN Consortium thus not only covers the questions of "What is it?" and "What does it do?", but also "What is it used for?"

Government agencies

Several government agencies throughout the world rely on systems engineering for purchases within defense and infrastructure and have defined architecture frameworks for description of system architecture to facilitate the procurement processes. Among the best known of such architecture frameworks are the Department of Defense Architecture Framework (DoDAF), Ministry of Defense Architecture Framework (MoDAF) and Atelier de Gestion de l'Architecture des Systèmes d'Information et de Communication (AGATE) developed by the US Department of Defense, The UK ministry of Defense, and the French Government Defense procurement agency respectively. All of these frameworks offer similar definitions of architecture that equate architecture to the principle partitative structure of a product, however from the content of the three architecture frameworks it is demonstrated that while the architecture definition is focused on the functional and partitative component structures of the system, the architecture phenomenon covers a broader range of subjects including non-structural aspects of the system. And it could be argued that the system architecture constitutes more than a structural description of a system. As an example AGATE defines architectural views for systems and systems of systems covering:

- Stakes and objectives of the system
- Description of the related organizations
- Processes and information flows
- Security requirements, in compliance with DGA policy
- Services of the system, and traceability with operational needs
- Conceptual structure of the system
- Physical structure of the system
- Life-cycle of the system

An AGATE model is organized into 5 views (Ministère de la Défense, 2005):

- View of Challenges, Objectives and elements of Context: This view describes synthetically, operational, strategic, economic, technical and financial elements of context targets, deadlines, supplies needed and risks associated with the proposed construction or development of the system.
- Business architecture: Describes the business relationships used to describe the specific relationships that underlie the functional organizations and business processes within the information system.
- Service-oriented architecture: Describes the services required by users and offered by the system.
- Logical architecture view: Describes the conceptual organization of the system. This representation ignores implementation choices set by the deployment or the products used to implement the system.
- Technical architecture view: Describes the implementation in hardware and software. The technical architecture is used to specify the range of selected products and how they are assembled into practice.

Note that there is an inherent understanding of an architecture as having multiple structures among them the so-called logical architecture and a physical architecture. This seems to suggest that there is an understanding of an architecture as consisting of multiple structures, among them the service, logical and physical architecture, and that any view of an architecture also includes views of non-physical elements. More importantly the architectural views defined by AGATE demonstrates that the system application is a part of describing the architecture, and could be argued that in the context of these frameworks, the architecture is not just concerned with the functional and part structure of the system, but also serves as a view of how the intended application and context of application of the system are addressed in the design of the system much in the same way as the OPEN Consortium sees it.

10.3 Conclusions on System Architecture

As with product architecture the definitions of what constitutes a System Architecture varies within systems engineering but shows a great similarity to views of product architecture. Within Systems Engineering there are those who define the architecture simply as the hardware and software structure of a system, while others offer a much broader definition such as those of ISO/IEC/IEEE42010:2011(E) standard and the OPEN Process Framework. The definition from ISO/IEC/IEEE 42010:2011(e) expands upon the restrictive definition of system architecture, by not limiting it to the functional and part structures of the system, and furthermore by characterizing the architecture as an expression of what constitutes the essential aspects of a system. The AGATE framework offers a good demonstration of how the architecture phenomenon covers not only the structural aspects of a system, but also other key elements which characterize the system including the intended application of the system as exemplified in three of the views in AGATE:

1. View of Challenges, Objectives and elements of Context
2. Business architecture
3. Service-oriented architecture

(Golden, 2013) further supports the application related aspect of architecture by stating that the architecture is a description not just of the technical, but of decisions and rationales i.e. the connection between application and design. I would suggest that this must be included in a definition of production system architecture. From the different perceptions of architecture within systems engineering it can be concluded in general that:

- Architecture is inherent to all systems, and there is a difference between the architecture of a system and an architecture description.
- The same constitutive and functional behavioral phenomena seen in product design are covered by the architecture phenomenon in systems engineering.
- The architecture concept can be seen to cover more than the structural aspects of systems. It also addresses the intended applications of the system including the business related roles of the system.

11 Towards an understanding of Production System Architecture

The review of current definitions of product architecture and system architecture demonstrates that while there does not exist one common definition or perception of what constitutes architecture for products or other engineered systems, the wide range of existing definitions provide a good foundation for a working definition of production system architecture. There is however a need to expand upon the concepts and to provide a better explanation of the architecture phenomenon in relation to production system design. This section seeks to provide a definition and an understanding of the architecture concept in relation to production system design based on the contributions from the disciplines of product design and systems engineering. In order to arrive at a working definition of production system architecture, contributions from the two disciplines are merged, reconciled and expanded upon as needed.

11.1 A definition of production system architecture

ISO/IEC/IEEE 42010:2010(e) provides the basis for the understanding of an architecture that is applied to production systems in this research. In accordance with the standard a production system architecture is considered to be an inherent aspect of any system of interest, meaning that a system of Interest can be said to exhibit architecture. The architecture of a system of interest can be described by an Architecture description which is said to express architecture (see Figure 11). In this way the distinction between a production system architecture and its description is explicitly stated.

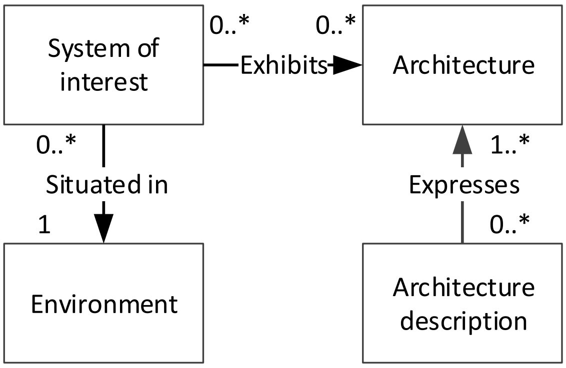


Figure 11 - Context of architecture and architecture description. Redrawn excerpt of figure 1 from (ISO, IEC and IEEE, 2011, p.3)

The architecture definition by ISO/IEC/IEEE 42010 probably provides one of the best foundations for understanding production system architecture, because while it is still largely structurally oriented, it is not as limited as many other definitions. This definition is more focused on the architecture as the conceptual container of the key aspects and characteristics of the system, and does not merely see the architecture as a structural reference model for the system. The definition of production system architecture in this dissertation is therefore adapted from ISO/IEC/IEEE 42010:2010(E) and the architecture of a production system is defined as:

> Fundamental concepts or properties of a production system embodied in its elements, relationships, and in the principles of the system's design and evolution that address the requirements and constraints from its intended applications.

It is also stated that:

> The architecture of a production system constitutes what is essential about that system considered in relation to its environment. There is no single characterization of what is essential or fundamental to a production system; that characterization could pertain to any or all of:
>
> a) system constituents or elements;
> b) how system elements are arranged or interrelated;
> c) principles of the system's organization or design;
> d) principles governing the evolution of the system over its life cycle; and
> e) the relationship between the system's constituent design and the intended applications or roles of the system in its environment.

The production system architecture definition provides a broad definition of what elements and relations are included in the content of an architecture. In the case of production systems these system elements and their relations are to be found in the elements of a transformation system as defined in the theory of technical systems. It should be said however that while the definition of production system architecture does not limit the architecture's constituent elements, product design theories do provide a clear indication that there are certain elements and relations that are of special importance in the design of products or production systems. These are: processes (also referred to as activities or functions), organs and parts (physical and non-physical); and their respective constitutive relations and allocation relations.

11.2 A layered framework for understanding the architecture phenomenon

One of the most important changes to the architecture definition compared to ISO/IEC/IEEE 42010 lies in the view that architecture can be seen as an expression of the principles of a production system's design and evolution to address the requirements from its intended applications or roles within its environment. This definition frames the architecture not simply as an expression of how functionality is achieved through the design of the system, but more broadly it frames architecture as an expression of how the different applications or roles of the production system are addressed in the design of the production system.

The basis for this view of architecture lies in the acknowledgement of the mutual relations between the production system and the functional areas of a company. It has long since been recognized that the production system is a key enabler of strategy in a company and that the design of production systems must not only address processing related issues such as productivity and quality (Skinner, 1969). This principle of a connection between the architecture and the functional areas of a company is similarly exemplified by (Yassine and Wissmann, 2007) who explore the phenomenon from a product design perspective, and show how product architecture relates to key areas of a company with mutual requirements and constraints as a consequence. The conclusion that can be drawn states that the application or role of the production system in the company reaches beyond the production function of the company. This means that the design of the system must not only address the functionality in the utilization stage associated with the production of products, but instead it must address all the different applications or roles that the system has within the company. It is therefore crucial to consider the production architecture phenomenon not just in the context of the composition of the constituent artefacts and functionality of a production system i.e. "What is it?" and "What does it do?", but also in the context of the "What is it used for?" To put it another way, the relation between the production system design and function of the company, means that the requirements and constraints of the design are not only found in the technical specification of the intended realized Transformation process. The requirements and constraints are also found in corporate strategies such as the Product strategy, Technology strategy, Sourcing strategy etc., as well as the governing doctrines, and applied tactics and specific operations of a company. When this view of the system design is taken, the architecture of the system can be seen not only as an expression of the systems structural composition and the scheme by which functionality is mapped to the physical elements of the system as defined by (Ulrich, 1995), the architecture becomes the scheme by which the applications of the system are realized through the design of the system. This is expressed in the conceptual framework of Figure 12, which provides a conceptual understanding of the production system architecture as a phenomenon consisting of an Application layer and Design layer.

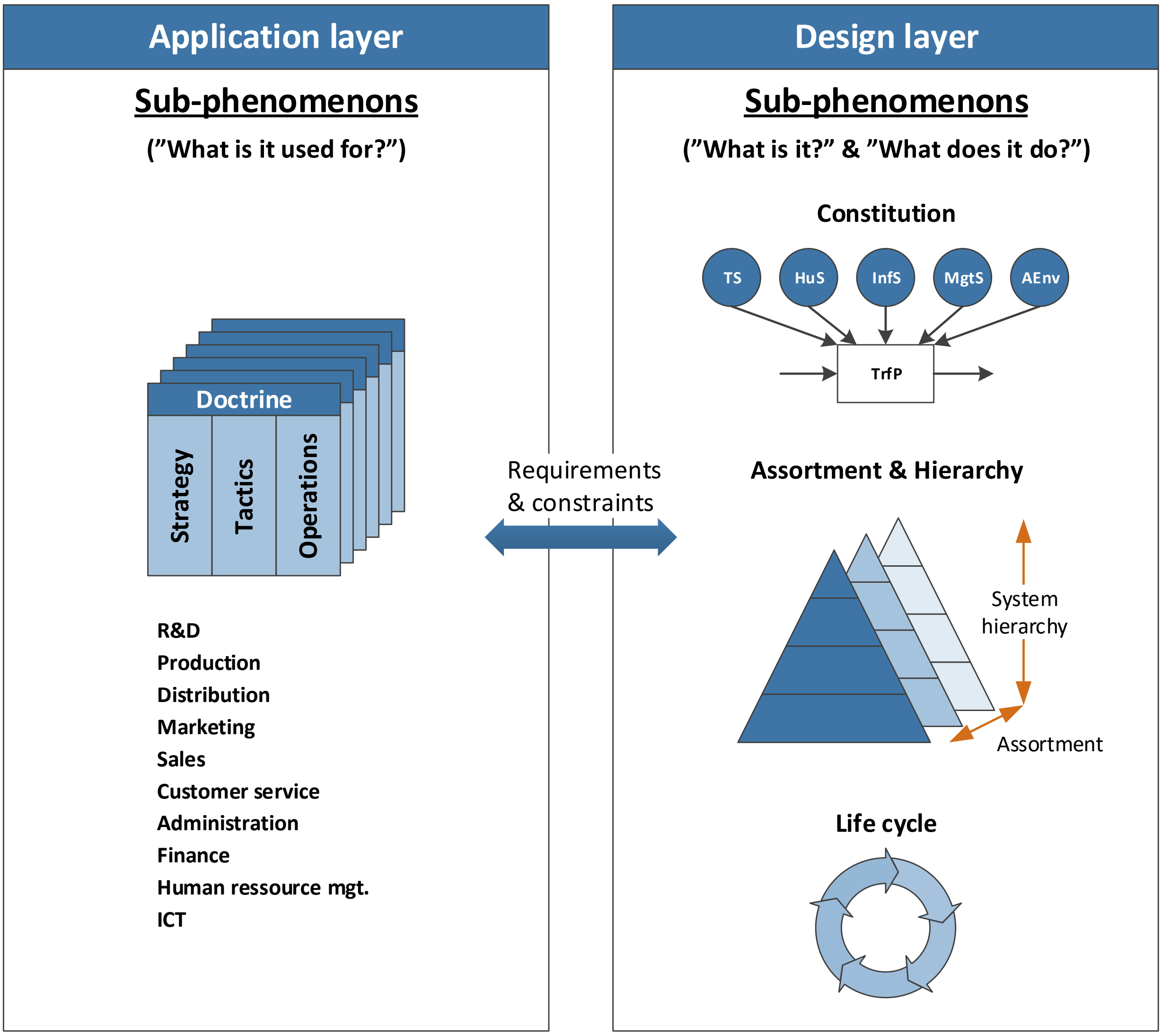


Figure 12 - Conceptual framework for the production system architecture phenomenon

Design layer: The architecture design layer describes phenomena associated with the constituent design of the production system. In other words the design layer can be said to answer the questions "What is it?" and "What does it do?" Among the sub-phenomena covered by this layer of the architecture are assortment and life-cycle phenomena that describe the sharing aspects between multiple systems and the changeable aspects of systems demonstrated in their life-cycle.

Application layer: A production system has several intended or unintended applications or roles to fulfill within the company, only one of which is the application of the system for production of products. The architecture application layer describes the phenomena related to the applications or roles the system fulfills. These phenomena are related to the doctrines, strategies, tactics and operations of the company in which the production system is conceived, designed, built and operated. The application phenomena provide the requirements and constraints which determine the principles of the systems organization and design as described in the architecture definition. In other words phenomena in the application layer can be said to answer the question "What is it used for?" The answer to this question provides the input for the design.

Linking the two layers

On their own the two layers of the architecture describe the applications and the design of the production system. In observing the link between the two layers the architecture helps to explain the relations between the applications and design, e.g.:

- Is the production system design consistent with the prevailing doctrines of the company?
- How does the production system design support or constrain the various company strategies, tactics and operations?
- What design requirements are imposed on the production system by the product strategy, finance strategy, etc.?

The link between applications and design is expressed by mutual requirements and constraints. The applications of the production system provide the requirements and constraints which govern the design of the production system and determine the specific characteristics of the system. Conversely the production system design will provide requirements and constraints for the applications of the production system. An obvious example would be to say that the product strategy will determine many of the key requirements for the production system, and that the production system will constrain the possible product strategy of the company. The architecture in this way also explains how a production systems design affects the rest of the company through feed back into/influence of the doctrines, strategies, tactics and operations.

There is no single answer to what applications will be relevant for any given production system, it could be everything from the Production Technology R&D to Organization. The architecture of any given production system will express the relationship between each relevant application and the design of the production system. It is not the subject of this research to explore the full extent to which production system design enables, realizes or constrains the doctrines, strategies, tactics and operations associated with different company functions.

The two layers of the production system architecture framework, and the included sub-phenomena, will be described in section 12 and 13.

12 The architecture application layer

Production systems play a key role in the realization of many of a company's strategies, and shape the possible tactics used in the company's operation. The application layer describes the applications and roles fulfilled by the production system within the company i.e. how the production system relates to the relevant doctrines, strategies, tactics and operations of a company including the resources involved in the formulation and/or execution of these. In this sense the application layer shows that the production system architecture is not simply the scheme by which functionality is implemented through the elements of a production system, it is also the scheme by which relevant doctrine, strategy, tactics and operation is realized through the system design e.g. the scheme by which the product strategy is realized in part through the design of a company's production systems. To properly understand this aspect of the architecture phenomenon, the following sections will provide an explanation of the concepts of doctrine, strategy, tactics and operations and how they can be categorized.

Doctrine, Strategy, Tactic and Operations

It is a common fallacy among people who talk of strategy, tactics and operation, to equate the three concepts with levels of organization and planning. Although organization and planning may center on these concepts, they are not to be directly equated with one another. Additionally it is uncommon to see people speaking of doctrine in a business context, but awareness of the concept can bring a better understanding of the assertions underlying a company´s goals, organization, operation etc., all of which have relevance for research into production system design. (Rao, 2007) offers four simple definitions for the four concepts:

- *"Doctrine: Doctrine is the set of assertions we accept as true in an action domain.*
- *Strategy: A strategy is a set of action and sequencing commitments, consistent with doctrine, and driven by the unique features of an action domain that constrain, but do not define, plans and schedules.*
- *Tactic: A tactic is an abstract action that can be applied in any of a large class of situations that conform to set criteria.*
- *Operations: Operations is the discipline of realizing strategy in the context of a background of infrastructure systems, resources and processes using a vocabulary of tactics."*

(Rao, 2007)

Since the various strategies, tactics and operations of a company are not the focus of this research beyond architecture-centric design, no further explanation of the subjects will be given. However a short detailing of doctrine does seem in order, since this is not always addressed together with strategy, tactics and operations in a business context. The concept of doctrine is most frequently used within military theory, but also has a role to play in a business context. In military use doctrine constitutes an approach to warfare, a sort of common reference frame for those engaged in military planning and operation. Doctrine such as this can be both explicit and tacit. Examples of explicit doctrines include the well-known tactical doctrines of Trench warfare and Blitzkrieg employed in the First World War and Second World War respectively. Within a business setting doctrines often go by other names or are not explicitly described. Instead doctrine exists as unspoken common understandings or approaches to the different tasks of the company, a sort of commonly agreed upon way of accomplishing the tasks of business. Other times doctrine is explicitly described, featuring both descriptions of the underlying principles and sets of specific tactics employed. Explicitly described doctrine can typically be found in internal company procedures for key company functions e.g. Product development, Technology development and Production system development. These doctrines describe principles, models, processes and tools used in key functions of the company such as Product R&D, Production R&D and Sales & Marketing. Very often these will include variations of different commonly known doctrines such as Participatory design, Open Innovation or Six Sigma. Business doctrines come in three forms: Discipline specific, Domain specific and Company specific (see Table 1):

Table 1 - Types of doctrine

Type	Explanation	Example
Discipline specific	Doctrine which is specific to certain types of tasks and can be applied to different domains/functions of the company	User-centered design Participatory design Open Innovation Six Sigma
Domain specific	Doctrine which is specific to domains/functions in the company. These can be applications of discipline specific doctrines to particular domains	Lean product development Lean software development Lean manufacturing Concurrent engineering
Company specific	Doctrine which is specific to a company, based on its particular functions, resources or organization. Such doctrine is either completely unique or based on elements of domain or discipline specific doctrines.	(These doctrines are often given unique names not used broadly in industry)

Categorization of applications

The categorization of the applications associated with doctrine, strategy, tactics and operations covered by the application layer can be done in many different ways, and depends on how these subjects are approached within the company. It is suggested that the production system applications should be viewed based on company functions, since this is frequently how strategy, tactics and operations are formulated. Examples of company functions could be:

- Research & Development (R&D)
- Production
- Distribution
- Marketing
- Sales
- Customer service
- Administration
- Finance
- Human resources
- Information & Communications Technology (ICT)

A company may view doctrine, strategy, tactics and operations in a different context than company functions, or they may apply to more than one company function at once. For the sake of simplicity the application layer of the architecture phenomenon is shown as covering applications for company functions. The important thing to note is that the architecture phenomenon also describes the relation between the production system design and the various applications as defined in company doctrine, strategy, tactics and operations however these may be defined.

As an example of what applications a production system could address, and what must be captured as part of an architecture description, we can consider the primary case company. In the case of Grundfos, production system architectures were to be defined within ten assortments of production systems which were considered of crucial strategic importance for the company. The architectures were a part of a desired shift to a new development doctrine for architecture and platform based design of both products and production systems. It was the intent that the architectures were to support, enable or be governed by the following applications:

Research & Development (R&D):

Doctrines	• Architecture and platform based development doctrine for products and production systems
Strategies	• Product development strategy • Production system development strategy • Technology development strategy
Tactics	• Design tactics a) Design for sustainability e.g. reduced energy consumption and sustainable energy sources b) Re-usability and recycling of equipment/system components c) Design preparation and reuse in development d) Support communication with stakeholders: Transparency in solutions e) Identification and visualization of dependencies (e.g. customer value and production capabilities). • Knowledge management tactics a) Knowledge sharing and transfer b) Intellectual property management • Technology management tactics a) Technology overview & analysis
Operations	• Product roadmap

Production:

Doctrines	
Strategies	• Globalization strategy • Capacity scaling strategy
Tactics	• Changeability tactics (see section 0) a) Process flexibility b) Product flexibility c) Volume flexibility d) Expansion flexibility e) Production flexibility f) Failure flexibility • Lean production
Operations	• Production plans on a short term (18 month) basis and medium term (< 5year) basis.

Distribution: N/A

Marketing: N/A

Sales: N/A

Customer service: N/A

Administration: N/A

Finance:

Doctrines	n/a
Strategies	• Investment postponement strategy (through gradual capacity scaling) • Investment reduction strategy (through cheaper production system designs)
Tactics	• Economics of scale in supply chain processes (through increased procurement volume and shared production systems)
Operations	• Compliance with deadlines and budgets • Enable better production capacity vs. space ratio

Human resources:

Doctrines	n/a
Strategies	• Organizational expansion strategy • Development organization localization strategy (globalization of development) • Productivity strategy in development organization
Tactics	n/a
Operations	n/a

Information & Communications Technology (ICT):

Doctrines	n/a
Strategies	• Standardization strategy for equipment & data management
Tactics	• Production data acquisition
Operations	n/a

Naturally some applications will be seen as more important influences for the production system architectures. In the case company this was in particular the production system development strategy which dictated a reduction goal of average time-to-production of 50%; and the investment postponement strategy, which focused on investment postponement enabled by gradual production capacity increases.

For any company transitioning to or engaged in architecture and platform based design and development of production systems, it is beneficial to review the different applications relevant for the production system design, and determine mutual requirements and constraints between the application and the production system design. These applications must be at the heart of a developed production system's architecture.

13 The architecture design layer

The design layer covers the phenomena related to constituent design of a production system or group of production systems; their variety; and their changeability across the system life. The layer covers three sub-phenomena: Constitution, Assortment & Hierarchy and Life-cycle. These three sub-phenomena are the answer to the two questions "What is it?" and "What does it do?" The Constitution phenomenon describes what the constituent elements of the production system are and how they are interrelated. The Assortment and Hierarchy phenomenon describes the recursive design phenomenon for production systems. And the Life-cycle phenomenon describes the phenomenon of changeability for production systems throughout their lifecycle. The following three sections will describe each of the sub-phenomena

13.1 Constitution

The Constitution phenomenon describes what elements a production system consists of and how these elements are related to each other. A production system is defined as a type of transformation system in accordance with the theory of technical systems. A production system is therefore constituted by the entities which make up the transformation system:

- Technical System (TS)
- Human System (HuS)
- Information System (IS)
- Management System (MgtS)
- Active and reactive Environment (AEnv)
- Transformation process (TrfP)

The five operators consist of elements existing in different domains e.g. Organs in the Organ domain and Parts in the Part domain (see Figure 13). Some element types and domains may be the same for multiple operators, because each operator often can be considered a transformation system itself (Eder, 2011). Specific elements are not shared however.

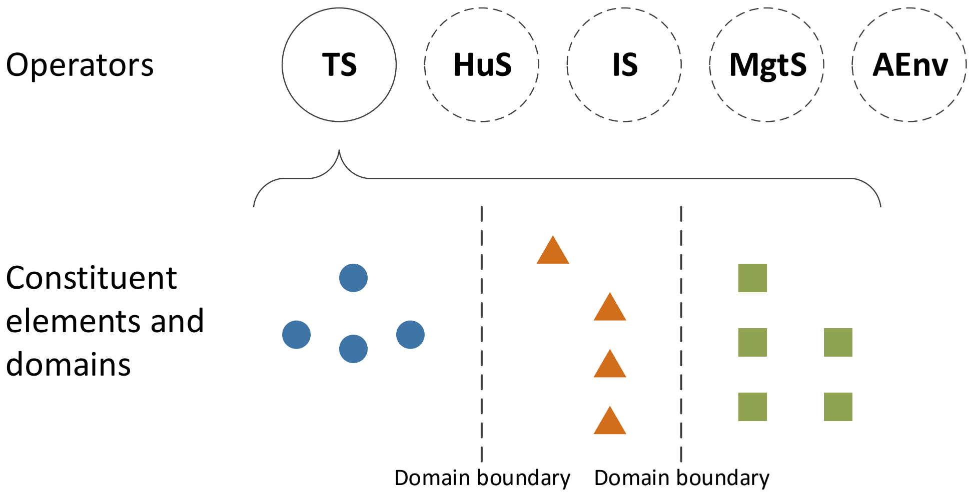


Figure 13 - Domain and elements for Operators

The elements of the different Operators are related to each other by two different kinds of domain relations: Intra-domain relations and Inter-domain relations (see Figure 14).

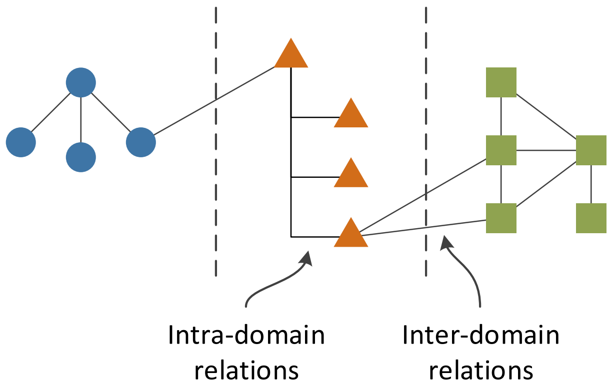


Figure 14 - Element relations

Intra-domain relations are relations between elements belonging to the same domain, and Inter-domain relations are relations between elements belonging to different domains. The Technical System provides an example of the constituent nature of one of the operators. In line with many of the architecture theories discussed earlier the Technical System can be said to consist of Organs and Parts, and be related to the transformation processes of the transformation domain (see Figure 15). The elements have intra-domain relations that make up some of the most important structures of the production system i.e. the Organs are related to each other in an Organ structure, and the Parts in a Part structure. Inter-domain relations are also present in the form of allocation relations between the three domains e.g. and allocations between parts and organs. In this way the constituent phenomenon describes how Transformations are implemented by Organs, and Organs are implemented by Parts.

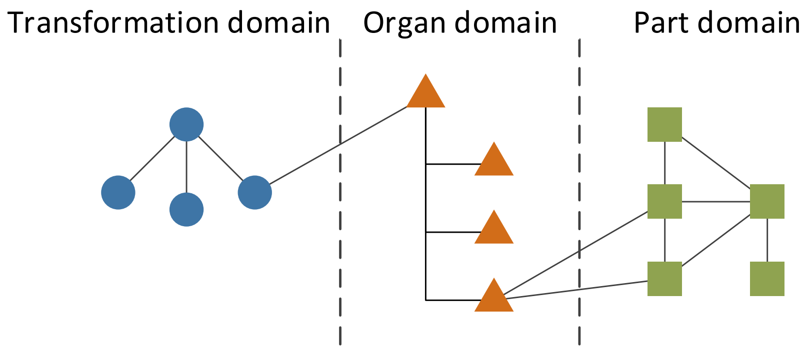


Figure 15 - Domains and relations for Technical Systems

Relations also exist between the elements of different operators. Parts for example have inter-domain relations to the physical spatial elements that are a part of Active and reactive environment, in the sense that Parts are located in Spaces. This is an example of a spatial allocation of Parts. The Spatial elements of the Active and reactive environment are themselves related to each other in a physical structuring of the spaces in the Active and reactive environment (see Figure 16).

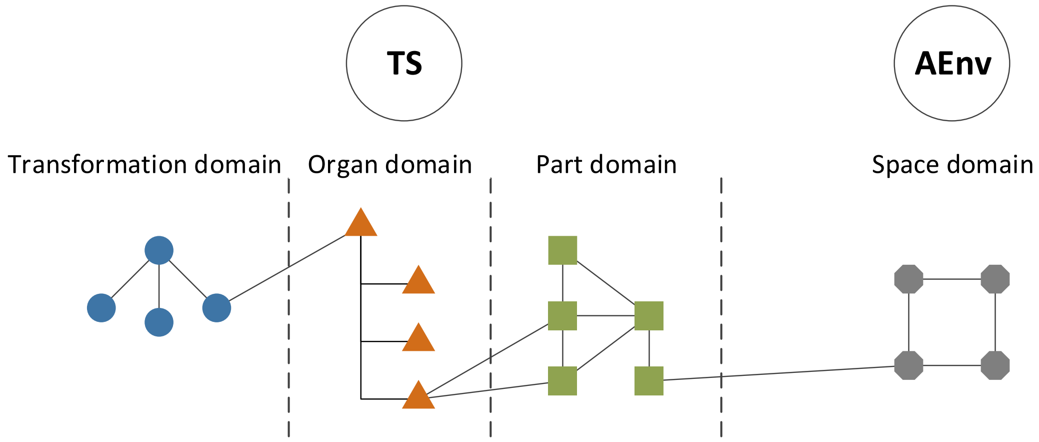


Figure 16 - Inter-domain relations between multiple Operators

Through the work at Grundfos and study of literature several different types of relations for the intra-domain and inter-domain relations have been identified, which describe different structures of the production system:

Composition (part-of): Composition relations describe the compositional relations between elements of the same domain e.g. the part structure of a production system, where parts consist of other parts. Note that higher level parts go by many different names e.g. modules, assemblies, sub-assemblies etc.

Attribute (kind-of): The attribute relation describes the relation between elements in terms of shared attributes. These relations are another way of saying that elements belong to the same class i.e. that they are the same kind. The relation can be viewed as a relation between elements and a meta- class that exemplifies the element type. Elements which have an attribute relation to each other are said to be the same kind or type of element. Attribute relations can be the basis of element classification. In the case of physical production equipment this classification is often expressed in industry standards for equipment, which specifies the type of equipment e.g. norm motors, electrical components, fittings and fasteners etc.

Interaction: Interaction relations are the relations that typically describe functional interaction between elements. Interactions between the Operators of the production system can be Material, Energy or Information (Eder and Hosnedl, 2007b). Some of the most important interaction relations in a production system are those existing between organs or parts i.e. the interfaces. (Pimmler and Eppinger, 1994) ads a spatial interaction and defines four classes of interfaces: Spatial, Energy, Information and materials; where the Spatial interaction identifies the relative spatial orientation between elements.

Allocation: Allocation relations describe different relations depending on the domains and element types. Between elements in the transformation domain and organ domain, the allocation relations describe how transformations are implemented by organs i.e. allocated to organs. And between elements of the part domain and elements in spatial domains, allocation relations describe how parts are allocated to spaces. If physical and non-physical parts are said to exist in different spaces, then the allocation relations can also describe how software is allocated to hardware.

Dependency: The dependency relation describes how the existence of an element is dependent on another element. This could mean that an element of one domain can determine if an element of another domain exists e.g. an Organ can dictate whether or not a specific part exists. An element can also influence the intra-domain relations of another object e.g. a transformation can dictate where in a part-structure a part appears, or it can dictate the specific characteristics of a part's interaction relations to other parts. These types of relations often describe configuration relations or requirements for the interaction between elements of another domain.

The different relations present in the architecture of a production system allow for description of many different structures depending on the viewpoint taken. This is further reason to support a broader view of architecture as more than the functional and physical structure of a production system. This is not to say that some structures are not more important than others, but the importance will be dependent on the viewpoint and the desired information to be obtained from describing the structures. In many cases the most important elements and relations are the Transformations (Processes), Organs and Parts which make up the Technical System and their respective Composition relations and Allocation relations.

It should be noted that the elements of the production system are not only related to each other but also to elements outside the production system i.e. the General Environment from the transformation system model. This is part of what is captured in the application layer of the architecture phenomenon. In this sense the application layer of the architecture phenomenon describes the relation between the company environment and the constituent elements and internal relations of the production system. The company environment includes the doctrines, strategies, tactics and operations incl. all related resources. Relations that connect system elements to the physical environment of the company, have to do with the operation applications e.g. how does the system relate to the existing production environment?

13.2 Assortment & hierarchy

The assortment & hierarchy phenomenon describes the relation between multiple similar or related production systems (assortment relation), and the relation between production systems of different resolution levels i.e. production systems existing in a part-of relationship (hierarchy relation). The following subsections will detail the two relationships.

Hierarchy - System levels

The hierarchy of production systems describes a compositional relationship between production systems, whereby one production system is a sub-element of another production system e.g. in the way that a machine can be a part of a production cell. The compositional relationship depends on the production system delimitation and the system characteristics on which this is based. Production systems can be decomposed based on different characteristics of the production system, and there is no right or wrong system breakdown, both in terms of the chosen characteristics for the breakdown and the specific system delimitation. (Wiendahl et al., 2007) describes the decomposition of factories based on two different views, a Resource view and a Space view. The resource view is based on the physical composition and operand transformed by the production systems, meaning that production systems which are involved with the transformation of the same operand are decomposed according to their physical composition. The Space view is based on a location based physical/geographic decomposition (see Figure 17).

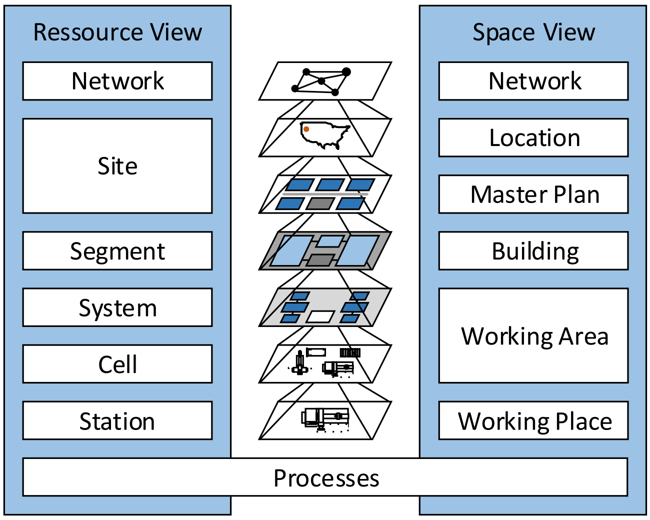


Figure 17 - Structuring levels and views of a factory (Wiendahl et al., 2007, p.785).

To precisely determine the characteristics of the production systems that are the basis of the system breakdown is not always easy, and so the system levels and delimitations are not necessarily fixed. Often a company will have its own classification scheme for production systems which provides a reference for definition of system levels and system delimitation. Such schemes may or may not be based on industry standards such as the process taxonomies found in DIN 8580:2003-09 (German Institute for Standardization, 2003), and they can often be found in a company's Enterprise Resource Planning systems.

Within this research production systems are decomposed based on a resource view of production systems and the systems under consideration cover those system levels where the design of the Technical system is the main focus. For higher level production systems the interest is often on the management and planning for production systems. Four levels of production systems are defined: Line level, Group level, Machine level and Machine element level (see Figure 18). These are equivalent to System, Cell, Station and Processes respectively from (Wiendahl et al., 2007).

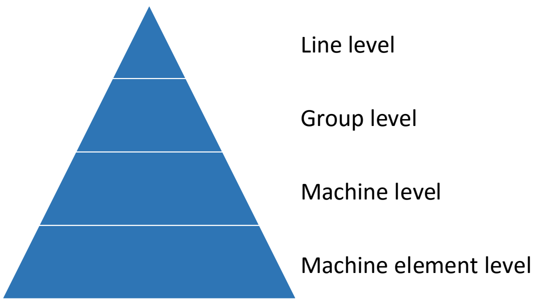


Figure 18 - Production system levels included in research

It can be very difficult to explicitly define delimitation of a production system at any of these levels, and the names are meant more as indications of what kinds of production systems would be found on a particular level. Descriptions of the four levels are given below and examples can be seen in Figure 19:

Line level: System/line/area (subject of production planning, typical lowest level of scheduling): an area of production system involved with the transformation of operands for the same product, product family or product assortment e.g. an assembly line.

Group level: The group level describes a collection of machines and machine elements responsible for carrying out more than one sub—process of the production process. The level is most frequently used to describe production cells e.g. an injection molding cell.

Machine level: Machine level production systems are equivalent to stand-alone production equipment often carrying out a sub-process of the production process. The level can also be considered equal to a station.

Machine element level: The machine element level describes production systems which are a part of a production machine e.g. tools, fixtures, grippers etc. Machine elements are not capable of completing a transformation process on their own. Only through the interaction of multiple machine elements can a transformation process be carried out.

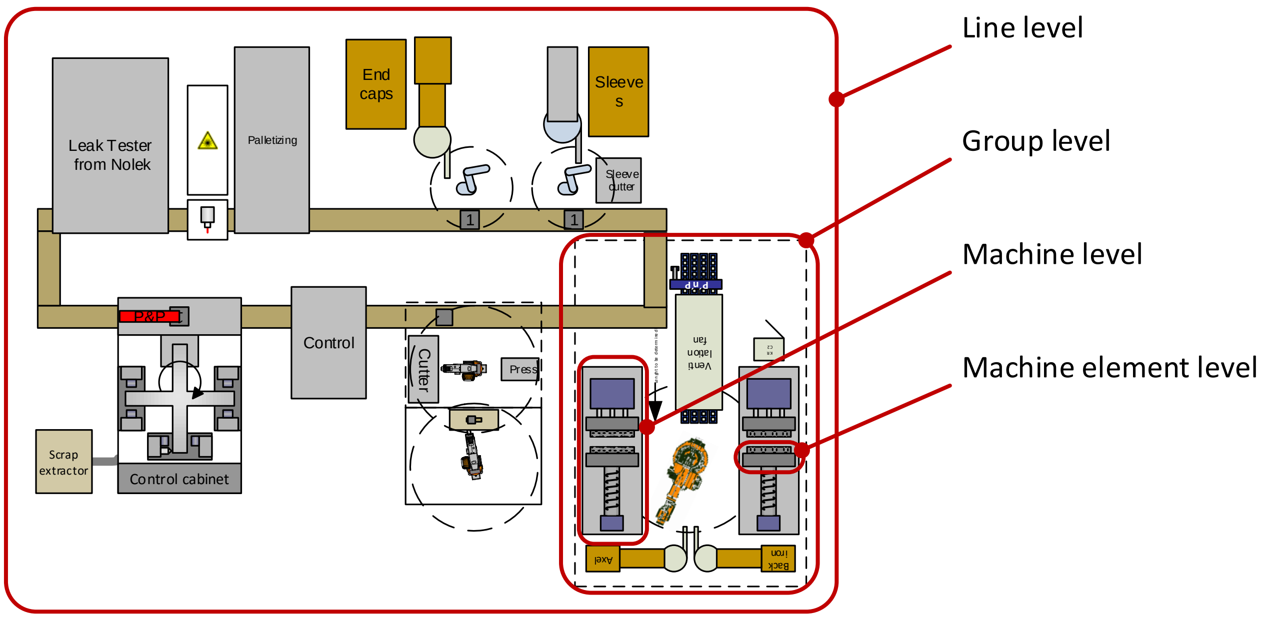


Figure 19 - Levels of production systems.

Assortment

The assortment relation describes the commonality or sharing across multiple production systems (see Figure 20). Assortments of production systems can be defined for the different levels of production systems defined in the hierarchies of production systems. The assortment phenomenon is well known from product design. (Harlou, 2006) describes assortments in relation to products and defines three levels: Products, Product families and Product assortment. There is no fixed number of levels to the assortment phenomenon, since more finely grained assortment definitions may be in play at different companies. It is reasonable to define similar assortment levels for production systems as for products i.e. Production system assortment, Production system family and Production system.

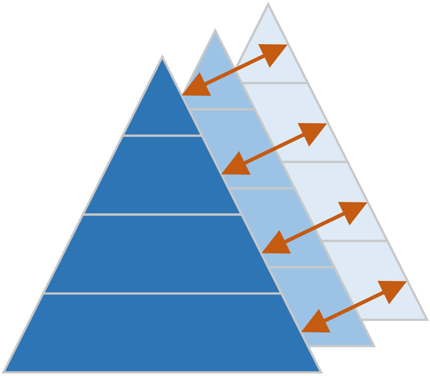


Figure 20 - The assortment phenomenon showing commonality across production systems.

The definitions of specific production system assortments, production system families and production systems are based on one or more aspects of the production systems. In this way the definitions will be equal to a classification scheme which is used to group the different production systems. Product assortments are most frequently defined based on a classification according to the marketing or sales definitions of products. In production this is not necessarily the best way to define assortments and there will be differences in the assortment definitions from one company to another. Examples include:

- Process technology
- Operand type
- Product
- Production location
- Automation levels
- Operator cost levels

It is not necessarily the case that a company uses the same aspect of production systems to define all the assortments of the company's production systems. Because the stakeholders have different needs for assortment definitions, some assortments are defined from an engineering design perspective, while others are defined based on a purchasing perspective, and yet others may be defined based on a financing perspective. It is not unusual for a company to use different aspects for assortment definitions for different levels of systems. Assortments for production lines for example may be defined based on the product to be produced, while machine assortments are defined based the process technology. Multiple definitions may also exist for the same production systems, so that it is defined as being a part of multiple different assortments of production systems. It would for example not be unusual to see the same machine associated with either:

- High automation/ low automation production systems
- Production systems for high cost/low cost production locations
- Technology specific assortments e.g., welding, laser welding.

Architectures exist both for individual production systems and for the different assortment groupings. (Harlou, 2006) defines architectures for products assortments and product families as consisting of design units that are either standard (called Standard designs) or non-standard i.e. shared or not shared between multiple production systems. An equivalent definition exists for architectures for production system assortments and production system families. Architectures which are shared/the same for multiple production systems are either assortment architectures or family architectures. These architectures describe the shared and non-shared aspects between the architectures of the different production systems. In cases where multiple production systems are built from the same architecture, the architecture is sometimes called a reference architecture.

13.3 Life-cycle

The life-cycle phenomenon is related to the life-cycle of the production system and the differences in the system composition that occur throughout. All production systems experience some degree of change throughout their life cycle, not least due to the prevailing trends in production. (ElMaraghy, 2005) names some of the major trends faced by production companies across the world:

- Shift towards mass customization
- Decreasing product life cycles
- Increased importance of delivery reliability over lead time and utilization
- Widening gaps between the life cycles of products, technology and production equipment
- Increased frequency of change in localization of production systems

The life cycle phenomenon within the production system architecture describes what the changeable aspects of a production system are, and how they relate to drivers of change that are shaped by the global trends for production. (Westkämper, 2006) identifies key internal and external factors that are the change drivers in production system design in the new global era (see Figure 21).

Figure 21 - Influencing factors for changeability (Westkämper, 2006).

The Internal factors (Human resources, Products/technologies, New methods, Networked structures) and External factors (Economy/Finances, Markets, Social/political factors, Environment) drive the Doctrines, Strategies, Tactics and Operations of the company as covered by the architecture application layer. The applications in turn drive requirements and constraints for the changeability of the production system. Because the applications of the production system are not limited to the production stage of the system´s life-cycle, the life-cycle phenomenon describes the changeable nature of production systems as it is seen throughout the entire life-cycle of the system. This includes not only the changes that can occur during the design and utilization of the system, but also the differences in the constituent elements and relations which are seen during earlier or later life-cycle stages e.g. during the construction/build of the production system.

The Life-cycle phenomenon explains how changeability objectives are addressed in the design of the systems constituent elements and their relations. This research project will not go into detail in regards to the specific changeability enablers for the different changeability objectives, since this is the domain of specific production design disciplines such as Modular design, Scalable design, Granular design etc. However it is one of the goals of the modeling contributions, to be able to express and communicate many of the listed flexibility in the description of production system architecture.

14 Conclusion on production system architecture

In Part 2 of the dissertation it has been established that production system architecture is a fundamental aspect of production systems, and that there is a difference between the architecture of a system and a description of the architecture. Architecture is broadly defined in accordance with ISO/IEC/IEEE 42010:2011(E) and not limited to the Technical System from the Theory of Technical Systems. The architecture phenomenon has been defined as a layered phenomenon consisting of an application layer, and a design layer. Where the application layer includes sub-phenomena describing the applications of the production system in relation to the doctrines, strategies, tactics and specific operations of a company incl. the related company resources. The conclusion form this, is that the architecture of a production system not only describes the constituent design elements and relations for a production system, but that more broadly speaking, the architecture is also an expression of the connection between the system applications and the constituent design. Analyzing the architecture of a production system can thus be used to answer questions such as how the design of a production system supports the company´s product strategy; how it enables specific development tactics; or how it fits to the organizational setup?

Based on the established definitions and understanding of the architecture phenomenon Part 3 will provide a contribution to the field of architecture description.

Part 3 – A contribution to description of production system architecture

Part 3 presents a contribution to the field of architecture description based on the principles of ISO/IEC/IEEE 42010. The contribution consists firstly of a conceptual model for architecture descriptions, that describes the constituent elements of an architecture description and their conceptual relations. Secondly a reference architecture framework is suggested, which can serve as the basis of architecture descriptions for production systems. The reference framework is referred to as the Production System Architecture Framework (PSAF) and it consists of viewpoints and model kinds that can be used in the description of production system architecture. Part 3 specifies the concerns and stakeholders that must be addressed by such a framework, and the tasks for which architecture descriptions based on the framework must be used. Specific viewpoints and model kinds that can be a part of the PSAF are detailed separately in Part 4 of the dissertation.

15 Architecture and architecture descriptions

It was established in Part 2 that there is a difference between the architecture of a production system and a description of such an architecture. Production system architecture was defined as a layered phenomenon that encapsulates several phenomena related to the systems constituent design and the relation between the design and application of the system. An architecture description on the other hand is a work product of the architecting process which expresses the architecture of a system of interest i.e. a production system (see Figure 22). According to (ISO, IEC and IEEE, 2011) the same system can exhibit several distinct architectures depending on the context in which it is considered, for example if it is considered in a different environment, where that environment includes developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences. This is the same as saying that the system would have a different architecture if there is a change in the application of the system. Similarly it is possible to express the same architecture through several different architecture descriptions depending on what aspects of the system are of covered by the description.

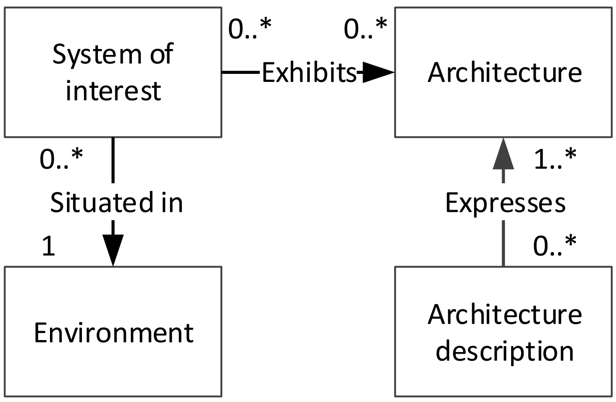


Figure 22 - Context of architecture and architecture description. Redrawn excerpt of figure 1 from (ISO, IEC and IEEE, 2011, p.3)

Architecture descriptions are created by system architects and used by many stakeholders throughout the life-cycle for different purposes, among other things for design and decision making support and documentation. Architecture descriptions take many different forms, from the simple to the detailed and the informal to formal (from simple hand drawings on paper to expansive computer models). Because architecture descriptions serve a variety of purposes many different conventions are used for their generation i.e. different kinds of models, notations and analysis techniques.

Various forms of architecture descriptions have been developed within architecture and platform focused research communities and within industry, although they may not always name them as such. Companies may very well employ architecture descriptions in their work, but not necessarily have a formalized approach to their basic definition, generation, use and development. Certain industries do have a more formal relationship to architecture-centric design and actively engage in the formulation and development of the basic concepts related to architectures as well as their description, development and use. Both generic and industry or application specific architecture related standards are published by organizations such as ISO, IEC, IEE and ASME. Government entities across the world also publish architecture related standards and guidelines for the purpose of supporting government procurement particularly within defense and infrastructure procurement.

In order to both take advantage of the existing standards within the field and in an attempt to address the problem of dissemination and adaptation of research in industry, the presented research contributions for architecture modeling and information handling will be based on and/or related to existing standards for development and use of architecture descriptions. Specifically the ISO/IEC/IEEE 42010:2011(E) standard for developing architecture descriptions will provide the structure for developing an architecture framework from which architecture descriptions can be generated.

16 A standard for architecture descriptions: ISO/IEC/IEEE 42010

The ISO/IEC/IEEE42010:2011(e) standard is a joint designation standard prepared by the Joint Technical Committee ISO/IEC JTC1 of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) in cooperation with the Institute of Electrical and Electronics Engineers (IEEE). The purpose and content of the standard is described as follows:

> *"This International Standard addresses the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions.*
>
> *This International Standard provides a core ontology for the description of architectures. The provisions of this International Standard serve to enforce desired properties of architecture descriptions. This International Standard also specifies provisions that enforce desired properties of architecture frameworks and architecture description languages (ADLs), in order to usefully support the development and use of architecture descriptions. This International Standard provides a basis on which to compare and integrate architecture frameworks and ADLs by providing a common ontology for specifying their contents."* (ISO, IEC and IEEE, 2011)

The standard specifies requirements for so-called *Architecture Descriptions*, but does not dictate specific content. Nor does the standard specify requirements for the described systems and their architectures or environments. As such it merely serves as the basis for describing the relation between production systems, production system architecture, and production system architecture descriptions (as seen in Figure 22). It also provides a useful context for development of the supporting modeling and information handling tools developed in this research, which can be used in architecture descriptions. The standard is intended for use by system architects as a part of their architecting activities, and is method neutral i.e. it is independent of the architecting method employed.

16.1 Elements of an architecture description

According to ISO/IEC/IEEE 42010:2011(E) an architecture description is intended to express the architecture of a system-of-interest and be used by the stakeholders who hold an interest in the system. The architecture description must therefore always identify the system-of-interest, its stakeholders, and their concerns. The system-of-interest could be a single production system or a group of production systems e.g. a family or assortment of production systems, and the stakeholders may be a specific individual, group or organization; or any type of these. In total architecture descriptions consist of five elements:

- Architecture rationale
- Architecture viewpoint incl. model kinds
- Architecture view incl. models
- Correspondence rule
- Correspondence

ISO/IEC/IEEE 42010 provides a conceptual model showing the relations between the five elements as defined by the standard (see Figure 23).

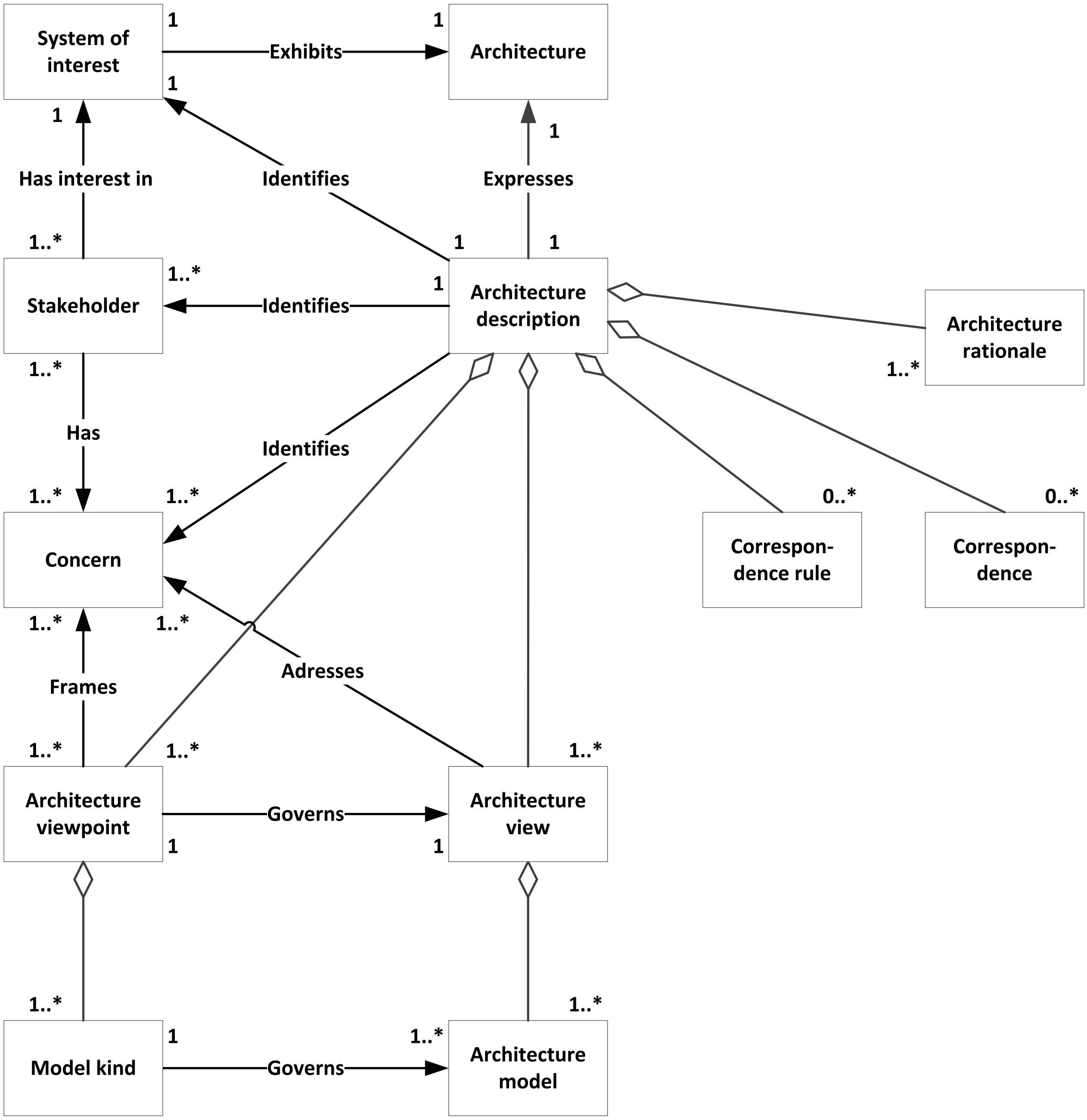


Figure 23 - Conceptual model of an architecture description. Redrawn from (ISO, IEC and IEEE, 2011, p.5)

At the heart of the architecture description are the *Architecture Views* consisting of different models. The views express aspects of the architecture related to the concerns of the stakeholders. The views are all based on a particular *Architecture Viewpoint*, which offers a particular way of viewing aspects of the architecture. The standard does not specify which specific viewpoints to use just as it does not specify the methods of architecting and creating architecture descriptions. The five elements of an architecture description defined by the standard are explained further in the following sections.

Architecture rationale

In ISO/IEC/IEEE 42010 *Architecture Rationales* are used for two purposes 1) recording the rationales behind *Architecture viewpoints* and 2) recording *Architecture Decisions*. The rationales for viewpoints explain why a certain viewpoint is used to describe the architecture, in relation to stakeholders, architecture concerns, model kinds, notations and methods used in the viewpoint. In the case of the viewpoints and model kinds developed or used in this research as part of the contribution to design decision support, the rationales will be given for each viewpoint. Architecture rationales for architecture decisions are used for documenting, explaining and justifying the underlying rationale for decisions regarding the architecture. The rationales identify which concerns and architecture description elements the decision pertains to or affects, and why and how (see Figure 24).

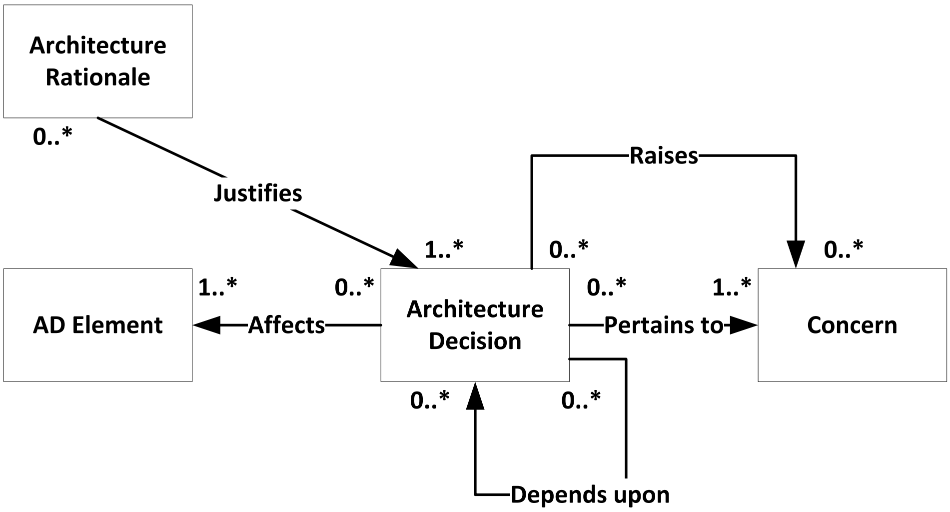


Figure 24 - Conceptual model of architecture decisions and rationale. Redrawn from (ISO, IEC and IEEE, 2011, p.8)

It should be noted that rationales for all decisions should not be recorded, since not all decisions are of equal importance. Only key decisions should have their rationales recorded.

Architecture views & viewpoints

Architectures for systems are typically described using many different kinds of models each covering different aspects of the system. These models are often gathered into cohesive collections of models that address related stakeholder concerns. These collections are called *Architecture Views* in the vocabulary of ISO/IEC/IEEE 42010, and they offer a way of viewing a systems architecture based on sets of related stakeholder concerns. As is well known, there is no agreement on what views or methods for expressing views are needed to properly describe a system's architecture. Different authors offer different views of system architecture to address the same stakeholder concerns, with different possibilities for understanding and analyzing the system's architecture. As an example (Andreasen, Hansen and Mortensen, 1995) describes four classes of views for expressing the structure of a product or product family, each addressing different architecture concerns (see Figure 25).

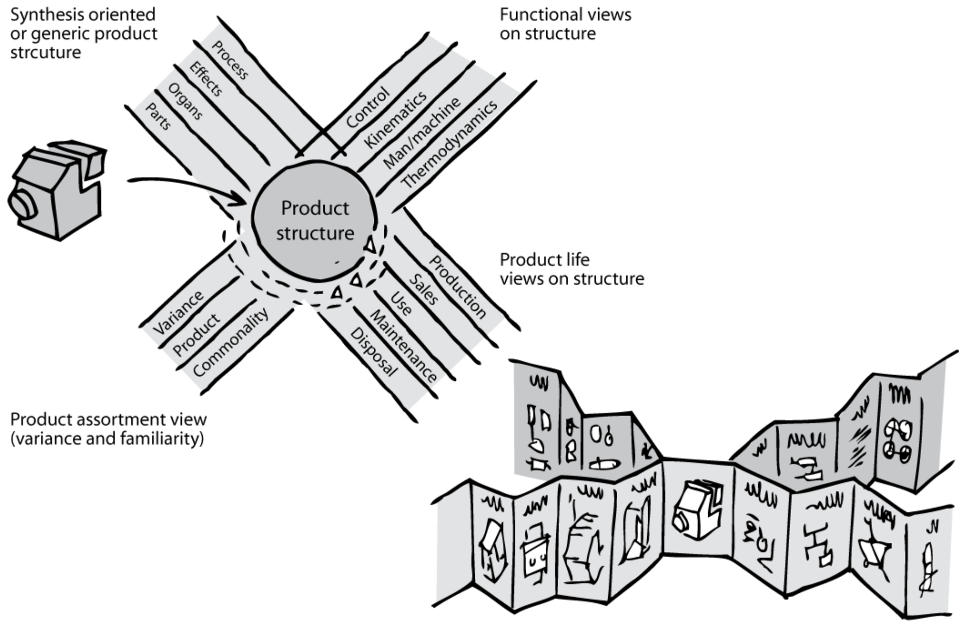


Figure 25 - The totality of product structure views (Andreasen, Hansen and Mortensen, 1995). Figure from (Kvist, 2010, p.43)

Views in architecture descriptions are governed by *Architecture Viewpoints* that define the types of models, methods and model correspondence rules used in the views. The viewpoints constitute a way of looking at a system with respect to a set of relevant concerns, and can be used for generating one or more views of the system. According to ISO/IEC/IEEE 42010 viewpoints specify:

> " a) *one or more concerns framed by this viewpoint[…];*
> b) *typical stakeholders for concerns framed by this viewpoint […];*
> c) *one or more model kinds used in this viewpoint;*
> d) *for each model kind identified in c), the languages, notations, conventions, modelling techniques, analytical methods and/or other operations to be used on models of this kind;*
> e) *reference to its sources."*
>
> (ISO, IEC and IEEE, 2011, pp.17–18)

Viewpoints can be defined either in the context of a specific architecture description, or independently of an architecture description. Viewpoints that are defined independently of a specific architecture description and can be used in multiple architecture descriptions, either directly or in an adapted form, are called *Library Viewpoints* (ISO, IEC and IEEE, 2011). These are the kind of viewpoints often described by researchers when presenting the uses of new modelling tools or methods. In this research project the contribution to design decision support includes the definition of library viewpoints that are used in collaboration between stakeholders from different disciplines to support decision making particularly in the early stages of the production system life-cycle. The viewpoints offer a view of the production system architecture that supports communication of key technical solutions and changeability aspects of the production system between stakeholders from different functions of the company.

Architecture models & model kinds

The architecture views which are used to express the architecture of a system consist of one or more *Architecture Models*, and these models could in turn be a part of more than one view, since they can be used to express parts of the architecture in relation to different concerns. There are many different perceptions of what constitutes such models, depending on who you ask. This research project does not limit what a model can be in the context of expressing an architecture, and will only describe modeling in more detail in relation to the modeling contribution described later in Part 4. The broad definition of a model by (Minsky, 1968) will therefore be applied:

> *"To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A."* (Minsky, 1968, p.425)

Models are governed by *Model Kinds* that encapsulate the conventions for models as expressed in the *"languages, notations, conventions, modelling techniques, analytical methods and/or other operations to be used"* (ISO, IEC and IEEE, 2011, p.18). Well known examples of model kinds include Design Structure Matrix, CAD models and UML diagrams. Just as models can be a part of different views, so will the governing model kinds be a part of the different viewpoints governing those views.

Correspondence & correspondence rules

Correspondence in architecture descriptions is used to express relations between the elements of the architecture description, and is governed by *Correspondence Rules*. The elements to which the correspondence relates may be any element contained within the architecture description e.g. Viewpoints, Views, Models, Model Kinds, Rationales or Decisions; or elements identified by the descriptions e.g. stakeholders, concerns or System-of-Interest. Of special interest is the correspondence relating to models, which allows for communication and interrelation of models i.e. model correspondence. Part 5 of this dissertation describes a contribution to model correspondence based on the ISO/IEC 81346 standard. The standard is used to develop a correspondence kind based on a reference designation system, which allows for communication of key constitutive elements and structures of the Technical system that can be found in many different model kinds. The contribution allows for interrelation of different model kinds, both those used in the viewpoints described in Part 4, and other state-of-the-art model kinds from the design research community such as Design Structure Matrix.

16.2 Architecture frameworks

It can be very beneficial to base architecture descriptions on a set of viewpoints and model kinds used by multiple system architects. (ISO, IEC and IEEE, 2011) introduces the concept of an *Architecture Framework*, consisting of predefined architecture viewpoints, model kinds and correspondence rules, from which architecture descriptions can be generated. According to (Hilliard, Malavolta, Muccini and Pelliccione, 2012, p.132) an architecture framework is *"a coordinated set of viewpoints, conventions, principles and practices for architecture description within a specific domain of application or community of stakeholders".* In short, an architecture framework constitutes a set of viewpoints including model kinds used for addressing a specific set of concerns, which can be applied to generate architecture descriptions for multiple systems-of-interest. According to (ISO, IEC and IEEE, 2011) architecture frameworks include the following:

“ a) information identifying the architecture framework;
a) the identification of one or more concerns;
b) the identification of one or more stakeholders having those concerns;
c) one or more architecture viewpoints that frame those concerns;
d) any correspondence rules.”

(ISO, IEC and IEEE, 2011, p.16)

Figure 26 shows the conceptual relations of architecture frameworks as defined by ISO/IEC/IEEE 42010.

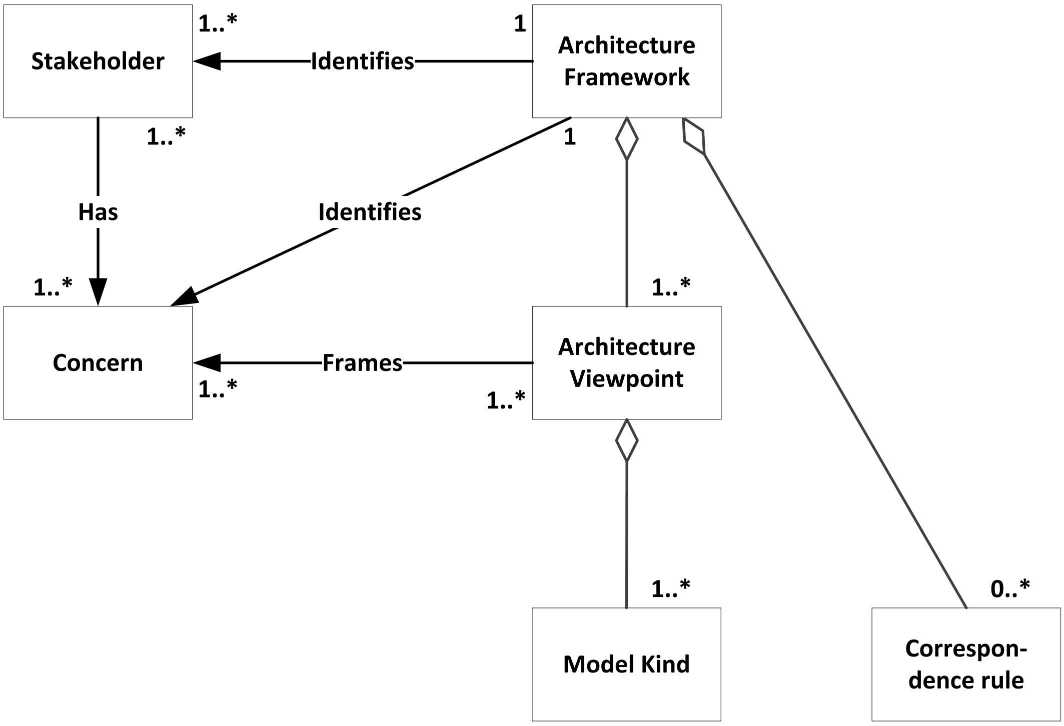


Figure 26 - Conceptual model of an architecture framework. Redrawn from (ISO, IEC and IEEE, 2011, p.10)

The purpose of defining architecture frameworks is to encourage reuse and interoperability within specific domains or communities of stakeholders. Such frameworks are most often defined within the domain of government procurement. Several government agencies throughout the world have defined architecture frameworks to facilitate the procurement processes related to defense and infrastructure. Among the best known of such architecture frameworks are the Department of Defense Architecture Framework (DoDAF), Ministry of Defense Architecture Framework (MoDAF) and Atelier de Gestion de l'Architecture des Systèmes d'Information et de Communication (AGATE) developed by the US Department of Defense, The UK ministry of Defense, and the French Government Defense procurement agency respectively. All of these frameworks offer their own definitions of what Viewpoints to use in an Architecture Description. For an overview of architecture frameworks for model-based systems engineering see (Reichwein and Paredis, 2011; Hilliard, 2014).

17 A revised conceptual model for architecture descriptions

In contrast to ISO/IEC/IEEE 42010 it is argued by (Emery and Hilliard, 2009) that every architecture description has contained within it an architecture framework. According to (Emery and Hilliard, 2009) the stakeholders, concerns, correspondence rules and viewpoints, which frame and govern an architecture description, constitute a framework specific to that architecture description. I believe this to be true and suggest a modified conceptual model for architecture descriptions (see Figure 27). The new model adjusts the model from ISO/IEC/IEEE 42010 by including an Architecture Framework and Correspondence Kinds.

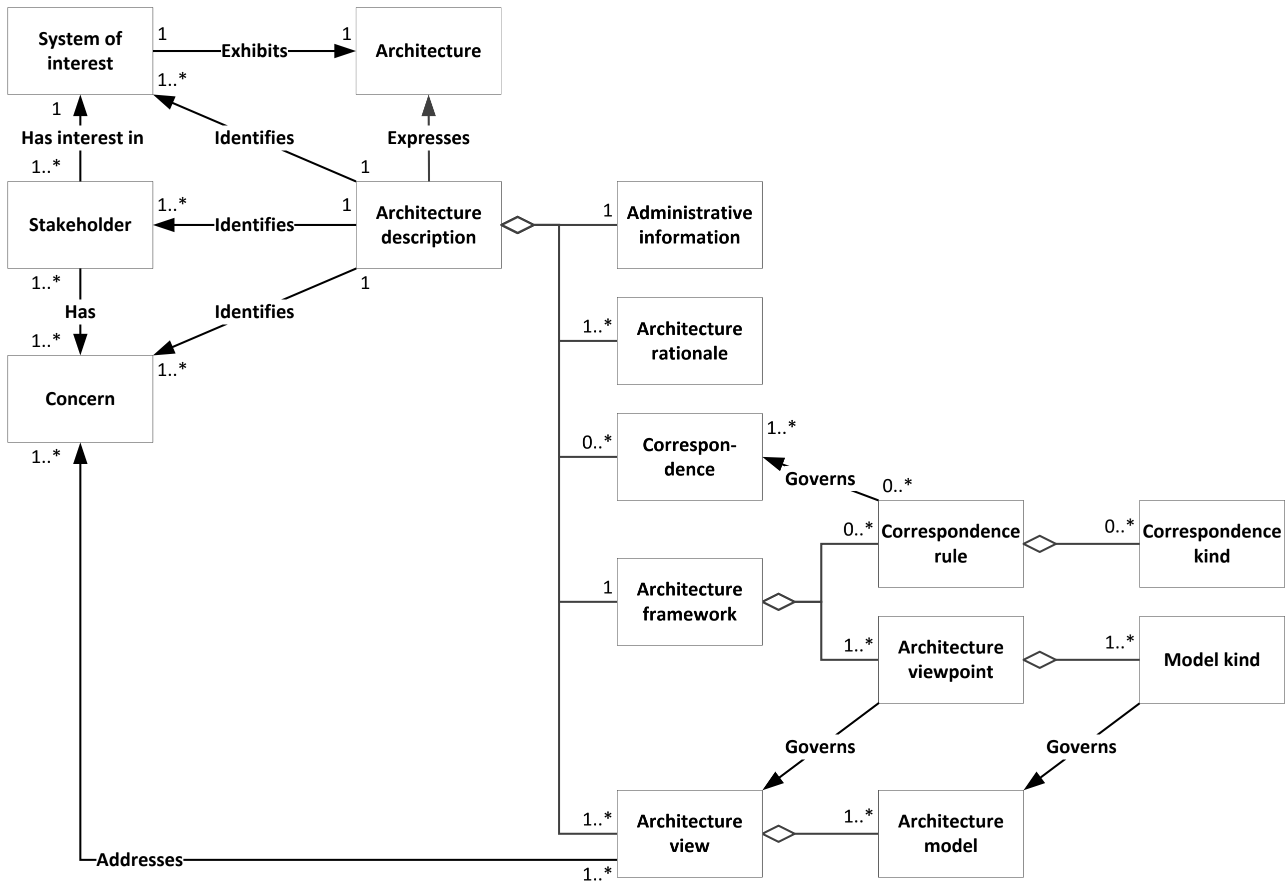


Figure 27 - New conceptual model for architecture descriptions

In the modified conceptual model an architecture description consists of five elements:

- Administrative Information;
- Architecture Rationales;
- Correspondence;
- Architecture Framework;
- Architecture Views.

The Architecture Framework in the modified conceptual model consists of:

- Architecture Viewpoints incl. Model Kinds
- Correspondence Rules incl. Correspondence Kinds

The Correspondence Kind element is a new addition to the content of architecture frameworks which specifies specific means of correspondence used in Architecture Correspondence as specified by the governing Correspondence rule. By specifying a correspondence kind capable of communicating key elements of the production system composition, and using this information in multiple models the design information can be more easily shared between stakeholders and across models.

In this new perspective on the relation between architecture descriptions and architecture frameworks, it should be noted that domain frameworks such as DoDAF and MoDAF are still architecture frameworks, but of a different nature to the framework of specific architecture descriptions. These frameworks are instead references which govern the makeup of the frameworks used in specific architecture descriptions. It is suggested to instead call these types of governing frameworks *Reference Architecture Frameworks*. An *Architecture Framework* is thus specific to the *Architecture Description*, but can be based on a *Reference Architecture Framework* that is commonly applied within the relevant domain e.g. DoDAF, which is applied within defense procurement in the USA. This means that the architecture frameworks for multiple architecture descriptions can be based on the same reference architecture framework (see Figure 28). The viewpoints used in reference architecture frameworks such as DoDAF are known as Library viewpoints.

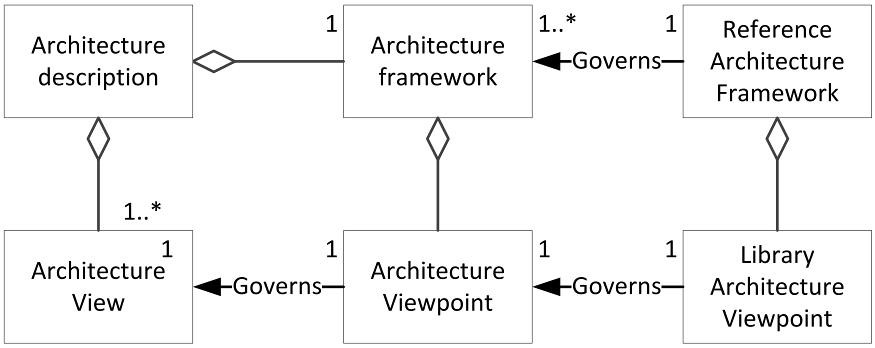


Figure 28 - Conceptual model for reference architecture framework

Currently there exists no Reference Architecture Framework for production systems used across industries and companies. Companies may define their own internal frameworks specifying certain models and information, which are generated in the design of production systems. These frameworks are often coupled to a process model specifying at what point in the development certain models are generated, by whom and for what purpose. This research project will develop the beginnings of a reference architecture framework for description of production system architectures, by contributing relevant Library Viewpoints and correspondence kinds to aid in the design of production systems. While this contribution is seen as the basis of a future framework used by multiple companies, the framework needs further testing and development before it can be directly applied by production companies at large.

The reference architecture framework will be referred to as the Production System Architecture Framework (PSAF). The remainder of Part 3 will be dedicated to describing the concerns and stakeholders that architecture descriptions based on the framework must address, as well as the possible uses of such architecture descriptions. Part 4 will introduce two library viewpoints contained in the framework, and Part 5 will introduce a correspondence kind contained in the framework, which can be used for model correspondence to support architecture information handling.

18 Developing a Production System Architecture Framework (PSAF)

The Production System Architecture Framework as it is described in this dissertation represents the starting point for developing a more formal description of the reference architecture framework. As such it could be considered to be a version 0.1. For PSAF to become a truly useful framework ready for use by practitioners there is a need for further development and testing, and for the framework to be documented separate from this dissertation in a manner that allows practitioners to apply the framework, including examples and usage guidelines. PSAF is as such not an official framework yet and is not described elsewhere than in this dissertation. In its current form PSAF is intended more as a starting point in the effort to consolidate some of the research contributions from this and other research projects at The Product Architecture Group at the Department of Mechanical Engineering at the Technical University of Denmark. It is suggested to use the framework to consolidate the research contributions from the research group, and aid in the dissemination of research contributions to industry. At this point in time this dissertation will therefore have to serve as the only identification of the framework under the name Production System Architecture Framework (PSAF or DTU-PSAF). This section specifies the necessary elements of a new reference architecture framework for architecture descriptions.

The reference architecture framework can serve as the basis for defining the architecture framework within the architecture description of a specific system. This means that many architecture frameworks can be based on the same reference architecture framework (see Figure 29).

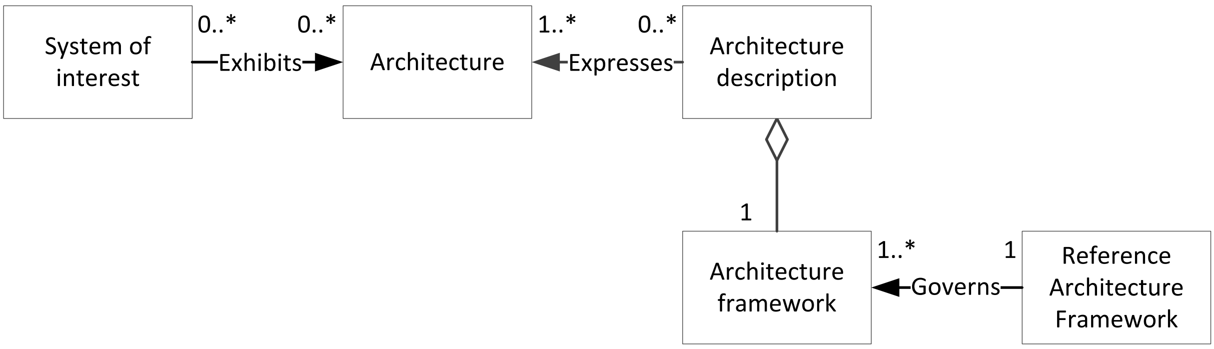


Figure 29 - Conceptual model for reference architecture framework

Following the new conceptual model for architecture descriptions (see Figure 27) from section 17, a reference architecture framework shall include:

a) Information identifying the reference architecture framework
b) Information regarding the use of the reference architecture framework (see section 19)
c) The identification of one or more architecture concerns (see section 20)
d) The identification of one or more stakeholders having those concerns
e) One or more library architecture viewpoints that frame those concerns (see Part 4)
f) Any architecture correspondence rules
g) Any architecture correspondence kinds (see Part 5)

The reference architecture framework can serve as the basis for creating architecture descriptions for specific production systems, but since the research focus is on the modeling part of the architecture description, certain correspondence and administrative aspects of the framework will not be fully developed in this dissertation. Specifically correspondence rules and identifying information will be omitted or described briefly, since the reference framework is not intended to be directly applied in industry without addition of these elements and further development and testing. Further testing of the reference architecture framework would also be necessary to establish the exact rules and guidelines for application of the framework.

Architecture descriptions based on PSAF must be usable for a multitude of stakeholders in the architecting process. The stakeholders who will hold an interest in any specific production system architecture will be specific to that system, but it is possible to define some typical stakeholders for the suggested reference architecture framework that determine which concerns the framework is meant to address, what viewpoints are needed, and what models are used. A stakeholder or stakeholder type can either be an individual (e.g. person, role or job), a group of individuals (e.g. interest group or unorganized network) or an organization (e.g. team, department, committee, organized network or company). Based on observations at Grundfos and from my work as a consultant the following stakeholder types are considered to represent the relevant stakeholders which need to use production system architecture descriptions in the design of production systems. The concerns of these stakeholders dictate the rationales behind the viewpoints and model kinds included in the reference architecture framework. These are the typical stakeholders which should be considered:

<u>Internal stakeholders</u>

- Individual
  - a) Engineers & systems architects
    - i) Systems engineers, system architect
    - ii) Procurement engineer
    - iii) Production engineer
    - iv) ICT engineers, ICT technician
    - v) Test engineer
    - vi) Quality assurance engineer
    - vii) Design engineers (e.g. mechanical, electrical, software)
    - viii) Compliance engineer
    - ix) Technology development engineer
    - x) Maintenance engineer
    - xi) Quality assurance engineer
  - b) Production planner
  - c) Equipment operator
  - d) Maintenance personnel
  - e) Documentation/data responsible (documents and manages ERP data)
  - f) Department managers & company directors
    - i) Project manager
    - ii) Product/assembly/component manager
    - iii) Production system development director
    - iv) Production manager
    - v) Technology development director

g) Project manager, new production system development

- Organization (department, team, network, committee)
  a) Production system development project team
  b) Production system engineering department(s)
  c) Equipment manufacturing department
  d) Technology development department(s)
  e) Human resources department
  f) Product development department(s)
  g) Equipment procurement department
  h) Quality assurance department
  i) Financial planning department
  j) ICT department (ICT)
  k) Production system development
  l) Logistics planning department
  m) Purchasing department

<u>External stakeholders</u>

- Individual
  a) Supplier engineers (sales and design)
  b) Supplier project manager
- Organization
  a) Equipment supplier (machine manufacturer, parts & equipment supplier)
  b) Materials/component suppliers

19 Uses of architecture descriptions based on a PSAF

Architecture descriptions have a wide range of potential uses in the creation, utilization and management of systems. The architecture descriptions that can be generated based on PSAF are not intended to support all such uses of production system architecture descriptions in a company, but will instead have a particular focus on supporting the creation of individual production systems and assortments of production systems. The architecture descriptions enable and support documentation and communication of the key system design aspects in order to provide a common understanding of the system design; and to provide a focus on the system architecture in activities relating to the production system's design and application. The focus in the prescribed library viewpoints and model kinds of the framework is on the main system design characteristics and how the system design addresses requirements related to the main design drivers found in the system applications e.g. system flexibility, variety creation, design reuse/sharing between systems, product strategy, finance strategy etc. The intention is to use the derived architecture descriptions to focus the design and development of the system on the architecture, and to use the architecture description as the center of communication between stakeholders both internally in the company and with external stakeholders. This particularly means to support communication between production system designers, utilization stakeholders, financial planners, and product and production system managers. Through this the architecture descriptions can become a point of understanding and agreement between stakeholders.

As expressed in the exploration of the production system architecture phenomenon, the production system design is influenced by and in turn influences a wide range of functions in a company. Production system architecture descriptions serve multiple purposes in these functions, some which directly serve to enable or support production, and other related functions which are influenced by or influences the production such as the finance or human resource functions. Understanding the potential roles and uses of the PSAF derived architecture descriptions can therefore be seen in relation to two main uses:

1. Roles and uses in the life-cycle processes of individual production systems
2. Roles and uses in all other company processes related to the various company functions.

The intended roles and uses of PSAF have primarily been compiled from observations made in the case companies through involvement in multiple development, procurement and management projects, as well as semi-structured and unstructured interviews with stakeholders from the companies. Additional relevant uses have been added based on ISO/IEC/IEEE42010. The collection of roles and uses make it clear that there is an almost endless list of uses for architecture descriptions. The presented roles and uses merely represent those which have been found particularly relevant in enabling and leveraging architecture-centric design of production systems. The roles and uses are explained in the next two sections.

19.1 Role and uses in life-cycle processes

Throughout its life-cycle a production system is the subject of a wide range of activities associated with the development, build, utilization, maintenance and disposal of the system. As such the range of what the architecture descriptions could describe in regards to the life-cycle of the production system is very wide. Some of the processes in the life-cycle in which the architecture descriptions are involved deal directly with the system, while other processes deal with the means by which the system life-cycle is realized. There are many different takes on what these processes are and this will differ from company to company. It is not the intention here to advocate for any specific process model to be used in defining the potential applications of PSAF, apart from specifying that a central focus on architecture throughout the life-cycle is a natural prerequisite. PSAF therefore is not required to be used in a specific process model. However for the sake of keeping to the systems engineering oriented view on the production system life-cycle, and because it is believed to offer a decent coverage of the most relevant processes in many companies, the systems engineering process model from ISO/IEC 15288:2008(E) is used to provide context for the role and uses of PSAF based architecture descriptions in the life-cycle. It is believed that the content and purposes of the processes included in the SE process model will also be included in the process models of most companies to a larger or lesser degree. The process model consists of four categories of processes (ISO and IEC, 2008):

- Agreement processes
- Organizational project-enabling processes
- Project processes
- Technical processes

Each of the four categories of processes consists of a number of specified sub-processes (see Figure 30). The processes can be modified or omitted to serve the specific purposes of an organization, and other processes deemed necessary by the organization can be included as well. The standard does not specify a specific progression of the processes, which is left up to the users. The role of PSAF derived architecture descriptions will be described in regards to the four categories of processes, but not for each sub-process.

Life-Cycle processes

Agreement Processes	Organizational Project-Enabling Processes	Project Processes	Technical Processes
Acquisition Process	Life-Cycle Model Management Process	Project Planning Process	Stakeholder Requirements Definition Process
Supply process	Infrastructure Management process	Project Assessment and Control Process	Requirements Analysis Process
	Project Portfolio Management Process	Decision Management Process	Architectural Design Process
	Human Ressource Management Process	Risk Management Process	Implementation Process
	Quality Management process	Configuration Management Process	Integration Process
		Information Management Process	Verification Process
		Measurement Process	Transition Process
			Validation Process
			Operation Process
			Maintenance Process
			Disposal process

Figure 30 - System life-cycle processes specified in ISO/IEC 15288:2008(E)

Technical Processes / Design processes

Technical processes are those processes in which the production system, seen both as a physical and a conceptual entity, is directly processed as the subject of creation, utilization and disposal. (ISO and IEC, 2008) defines the technical processes (where the 'product' is equal to the production system) as:

> *"The Technical Processes are used to define the requirements for a system, to transform the requirements into an effective product, to permit consistent reproduction of the product where necessary, to use the product to provide the required services, to sustain the provision of those services and to dispose of the product when it is retired from service."* (ISO and IEC, 2008, pp.35–36)

The technical processes are involved in all stages of the life-cycle, and architecture descriptions in general have a role to play in all of the technical processes. PSAF derived architecture descriptions however have a particular focus on support for the early stages of specification and design in the technical processes. PSAF derived architecture descriptions are to be used as a support for making the larger design decisions, and as a communications tools between stakeholders. The architecture descriptions are therefore specially aimed at enabling and supporting the design related processes of the technical processes, and have a smaller focus on the technical processes related to the later stages of system build, utilization, maintenance and disposal. The architecture descriptions thus help to answer research question two through support of the design related decision making and communication between key stakeholders. While the architecture descriptions do not focus on support for the later stages of utilization, maintenance and disposal of the system life-cycle, they can still serve as a reference for the system design in these stages and act as a means of documenting changes to the system and capturing knowledge relating to the utilization, maintenance or disposal of the system.

The main uses of architecture descriptions in relation to the technical processes are therefore as:

- Basis for…
  a) analysis and synthesis of system architecture
  b) definition of design sharing e.g. platform definitions or design principle definition
  c) communication of system architecture across stakeholders, processes and life-cycle stages
  d) verification of conformance by implemented production system to the architecture
  e) design feedback to the organization from later life-cycle stages
  f) knowledge capture through-out the life-cycle e.g. lessons learned, experience etc.
- Documentation of…
  a) technical system requirements and constraints
  b) system design, changeability, and processing & performance capability
  c) architecture decisions, rationales and implications
  d) design progression in life-cycle
  e) planned and realized system changes throughout the life-cycle

Project process activities

Project processes are facilitating processes used in the execution of the various projects concerned with the life-cycle of the production system. The project processes are defined as:

> *"…a set of non-engineering processes conducted within the range of responsibility of a project that need to be defined in order that system-specific technical processes can be conducted effectively"* (ISO and IEC, 2008, p.24)

> *"The Project Processes are used to establish and evolve project plans, to execute the project plans, to assess actual achievement and progress against the plans and to control execution of the project through to fulfillment."* (INCOSE, 2011, p.177)

Project processes have to do with project planning, assessment and control, and includes management processes relating to risk management, decision management, configuration management, information management and measurement.

Project processes can be divided into two categories: 1) Project management processes, and Project support processes. The roles and uses for architecture descriptions in these project processes are:

<u>Project Planning Process</u>

- Basis for...
  a) project planning
  b) definition of system related stakeholder roles and responsibilities
  c) resource planning
- Reference for and documentation of project progressions

<u>Project Support Processes</u>

- Basis for architecture related risk assessment
- Documentation of…
  a) system related elements subject to configuration management
  b) items subject to information management

Agreement processes

According to (ISO and IEC, 2008, p.15) agreement processes *"define the activities necessary to establish an agreement between two organizations"* for acquiring or supplying products or services in accordance with the acquirer's requirements. PSAF derived architecture descriptions can serve as a means of communication in this acquisition and supply relationship, and can also capture the organizational choices of suppliers. The roles and uses for architecture descriptions in these project processes are:

- Documentation of supplier selections
- Means of communication between acquirer and supplier

- Basis for preparation of acquisition documents e.g.
    a) Request for proposal
    b) Requirements specification and acceptance criteria for supplied product or service
- Reference for statement of work and organizational division of responsibility

Organizational project-enabling processes (1p)

The organizational project-enabling processes create the necessary capabilities of the organization required to enable the production system life-cycle.

> *"The Organizational Project-Enabling Processes ensure the organization's capability to acquire and supply products or services through the initiation, support and control of projects. They provide resources and infrastructure necessary to support projects and ensure the satisfaction of organizational objectives and established agreements. They are not intended to be a comprehensive set of business processes that enable strategic management of the organization's business."* (INCOSE, 2011, p.267)

In organizational project-enabling processes, the PSAF derived architecture descriptions, will serve as the input for decision making, planning and prioritization. The descriptions thus provide the support and reference the organization needs to provide the resources necessary for supporting the system life-cycle. Some of the roles and uses of the architecture descriptions would be:

- Basis for…
    a) defining, prioritizing and planning projects
    b) specifying requirements for infrastructure needed for the generation, utilization and maintenance of architecture descriptions
    c) specifying skill requirements for production system stakeholders e.g. engineering skills

19.2 Role and uses apart from life-cycle processes

The life-cycle processes are meant to describe the activities in a company directly related to the specification, design, procurement and implementation of a system including the enabling and supporting processes. This means that the life-cycle processes from (ISO and IEC, 2008) address a large part of the applications in the application layer of the architecture phenomenon, but they do not cover all the applications in the company that influence or are influenced by the production system design. This includes activities which support the entire organization or cut across projects. Some examples of these uses would be:

- Basis for…
    a) planning and scheduling of…
        i) production system design and development activities
        ii) product design and development activities
        iii) technology research and development activities
    b) organizational design and development

c) Knowledge management for production system design
d) Production system design preparation and reuse
e) Sourcing strategy for production equipment
f) Production system portfolio management
g) Investment strategy for production systems

20 Concerns framed by a PSAF

The PSAF derived architecture descriptions shall enable stakeholders to focus on the concerns of interest to them. This section describes the architecture related concerns of stakeholders that are framed by the PSAF derived architecture descriptions. The concerns all relate to how the production system applications are answered by the system design, as explained in the relation between the application and design layers of the architecture phenomenon.

In order to support the design and procurement of production systems, PSAF should help address the following stakeholder concerns:

- What can/should the system be capable of?
- What is the technical solution?
- How can/will the system change?
- What is the division of responsibility?
- How does the system fit into/relate to the assortment and/or hierarchy of production systems?

The concerns were captured through interviews, project observations (through participation), and a prolonged organizational presence (the researcher was physically placed at the organization 75% of the time the case lasted). The concerns are related to the concept/solution of the system, the capabilities of the system, the changeability of the system, and the supply of the system, and can be summed up into these four themes:

1. Processing capability
2. Constituent design
3. Changeability
4. Assortment & hierarchy relations

PSAF includes library viewpoints intended to frame the concerns relating to these themes which are deemed most relevant in the design and procurement stages of the production system life-cycle. The described stakeholder concerns may be held by many of the stakeholders of the system, and it is most probable that there will be additional concerns not covered by PSAF. When PSAF is used to generate a framework in the architecture description of a production system, the architecture description must document the association between the concerns framed by the viewpoints and the actual stakeholders of that particular system. The themes of concerns will be explained in the following sections.

20.1 Processing capability

The primary purpose of any production system is to carry out a production process or series of processes that output a finished product or work piece for further processing. Stakeholders are therefore especially concerned with the capabilities of the system in relation to the desired transformation process, and the possible performance of the system in carrying out the process. Please note that this capability may include the process capability (the statistical measure of process variability), but is not limited to this.

In general the processing capability is related to the effects of the operators on the operands e.g. effects of the technical system on the processed workpieces; the range of input and output of the system; and the processing capabilities of the system. Often stakeholders are very concerned with the functional behavioral aspects of the production systems, but this is not in focus for PSAF. PSAF instead offers a description of the operational capability in terms of the characteristics of the production process and the range of input and output of the system. In total the covered concerns relate to the following:

- Process flow and process flow variety
- System input/output range
- Work piece/product variety generation
- Part logistics to and from the system
- Production capacity

The architecture descriptions should be capable of describing how the system design can provide the desired capability, what the characteristics of the solution are, and how the solution is configured.

20.2 Constituent design

Stakeholder concerns relating to the constituent design of a production system have to do with the constituent make-up of the system. The PSAF derived architecture descriptions can be used for describing the make-up of the production system i.e. what elements the production system consists of and how these elements are inter-related. This includes describing the characteristics of the elements; relations between the elements such as part structures, interfaces and functional allocation. PSAF derived architecture descriptions are meant to describe only the Technical system in its current iteration. This means that the descriptions describe the elements of the technical system and the relations within the technical system and to the other operators of the production system, the production process and the general environment in which the production system is situated. With that in mind PSAF should frame the following concerns in relation to the technical systems:

- Technology selection
- Risk of solution
- System concept
  a) Elements, structures, allocations and configuration options
  b) Key attributes of system elements (e.g. Price, Capability, Size, Weight, etc.)
- Investment
  a) Price of system
  b) investment postponement options
- Constraints from production environment, requirements for environment

If PSAF is developed further in the future it should also address the constituent design of the other operators of the production system and the related concerns. There are many other concerns relating to the other operators which could also be of interest for the stakeholders such as for the human system the concerns could be in regards to the required operators for the system in the utilization stage, operator skill levels, necessary organization of the work force, requirements for versatility in the work and the operator's roles. For the information system, it would be relevant to know what the data collection in the system should be, how it happens, what the purpose of it is, how it should be treated, how it relates to the other operators e.g. how it influences the technical system and management system. For the Active and reactive environment it could be interesting to specify what requirements there are for the environment e.g. in terms of safety, noise, protection of workers, space requirements etc.

20.3 Changeability

Stakeholder concerns relating to the life-cycle of the production system often revolve around the changeable nature of systems in general and production systems in particular. The ability of a production system to change its make up or functionality depending on the requirements of any given life-cycle stage is one of the most important characteristics of a good production system design. The ability of the system to change to suit the different requirements encountered at different stages of the life-cycle is known as the system changeability, and a production system capable of change is known as a changeable production system or changeable manufacturing system. Specific types of changeable production systems may also be known by other names such as reconfigurable or flexible manufacturing system.

The changeability of the production system can be described in relation to a life-cycle model for the production system. Different authors describe life-cycle models on vastly different levels of detail, from a simple 3 stage model to a 22 stage model (Sage, 1992). The INCOSE systems engineering handbook describes seven generic life-cycle stages for systems in systems engineering, which is a sufficient understanding of the life cycle for the purposes of this research (see Table 2).

Table 2 - Generic life-cycle stages and their purpose (INCOSE, 2011, p.25)

Life-cycle stages	Purpose
Exploratory research	Identify stakeholders' needs Explore ideas and technologies
Concept	Refine stakeholders' needs Explore feasible concepts Propose viable solutions
Development	Refine system requirements Create solution description Build system Verify and validate system
Production	Produce systems Inspect and verify
Utilization	Operate system to satisfy users' needs
Support	Provide sustained system capability
Retirement	Store, archive, or dispose of the system

Regardless of how the life-cycles vary from system to system, the stakeholder concerns regarding system changeability can be roughly divided into two categories:

1. Changeability in the Utilization life-cycle stage (i.e. production)
2. Changeability in remaining life-cycle (i.e. differences in elements and structures across the other life-cycle stages.)

The two categories of changeability are explained in the next two sections.

Changeability in Utilization stage

The Utilization stage of the life-cycle is the stages where the changeable nature of a production system is most often of interest. Several types of production systems are defined in industry and academia which focus on the changeable aspects of a production system e.g. Agile manufacturing, Reconfigurable manufacturing and Flexible manufacturing. These different concepts describe different changeable aspects of production systems and can be commonly referred to as changeable manufacturing. Changeability in relation to production systems can be defined as:

> *"Changeability […] is defined as characteristics to accomplish early and foresighted adjustments of […] structures and processes on all levels to change impulses economically."*
>
> (Wiendahl et al., 2007, p.785)

Depending on which objects related to the production system are changed, different types of changeability for the production system design can be defined. (Wiendahl et al., 2007) defines five classes of changeability and designates their relevance in relation to the levels of production systems and products (see Figure 31).

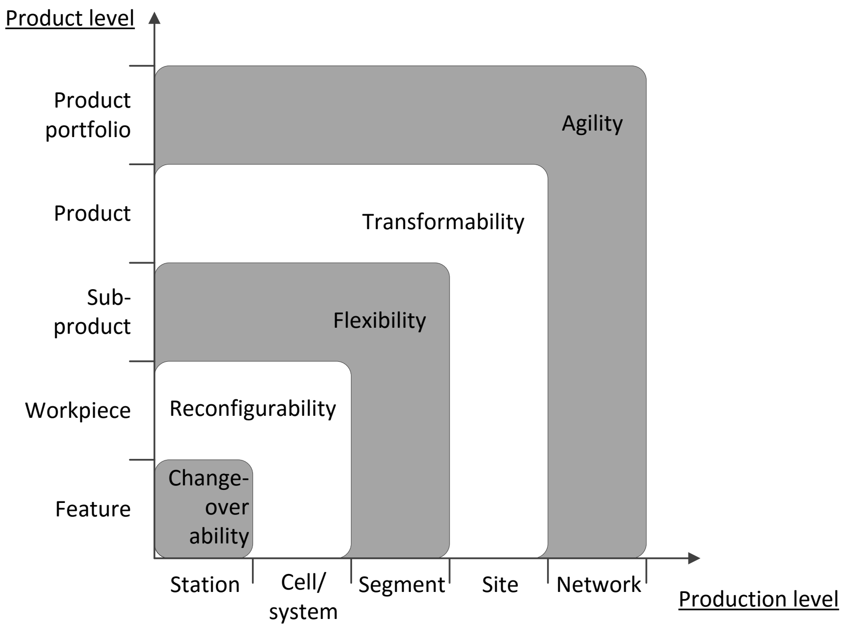


Figure 31 - Classes of production system changeability. Figure redrawn from (Wiendahl et al., 2007, p.786)

(Wiendahl et al., 2007) defines the changeability classes as:

> *“**Agility:** Agility means the strategic ability of an entire company to open up new markets, to develop the requisite products and services, and to build up necessary manufacturing capacity.*
>
> ***Transformability:** Transformability indicates the tactical ability of an entire factory structure to switch to another product family. This calls for structural interventions in the production and logistics systems, in the structure and facilities of the buildings, in the organization structure and process, and in the area of personnel.*
>
> ***Flexibility:** Flexibility refers to the tactical ability of an entire production and logistics area to switch with reasonably little time and effort to new – although similar – families of components by changing manufacturing processes, material flows and logistical functions.*
>
> ***Reconfigurability:** Reconfigurability describes the operative ability of a manufacturing or assembly system to switch with minimal effort and delay to a particular family of work pieces or subassemblies through the addition or removal of functional elements.*
>
> ***Changeoverability:** Changeover ability designates the operative ability of a single machine or workstation to perform particular operations on a known work piece or subassembly at any desired moment with minimal effort and delay.”*
>
> (Wiendahl et al., 2007, p.786)

The various classes of changeability have different changeability objectives and changeability enablers. Since this project only considers production systems at or below the Line level, not all changeability classes are of relevance. The changeability objectives for these production systems include changeability objectives associated with Changeoverability, Reconfigurability and Flexibility. Changeability objectives associated with Changeoverability and Reconfigurability are often included when objectives for Flexibility are defined. (ElMaraghy, 2005) identifies at least 10 types of production systems flexibility for production systems involved in parts manufacturing:

1. *“Machine flexibility: Various operations performed without set-up change.*
2. *Material handling flexibility: Number of used paths per total number of possible paths between all machines.*
3. *Operation Flexibility: Number of different processing plans available for part fabrication.*
4. *Process Flexibility: Set of part types that can be produced without major set-up changes, i.e. part-mix flexibility.*
5. *Product Flexibility: Ease (time and cost) of introducing products into an existing product mix.*
6. *Routing Flexibility: Number of feasible routes of all part types/Number of part types.*

7. *Volume Flexibility: The ability to vary production volume profitably within production capacity.*
8. *Expansion Flexibility: Ease (effort and cost) of augmenting capacity and/or capability, when needed, through physical changes to the system.*
9. *Control Program Flexibility: The ability of a system to run virtually uninterrupted (e.g. during the second and third shifts) due to the availability of intelligent machines and system control software.*
10. *Production Flexibility: Number of all part types that can be produced without adding major capital equipment.*"

(ElMaraghy, 2005, p.263)

PSAF seeks to be able to address as many of these changeabilities as possible in the description of the characteristics of the production process and the resources used in realizing the production process. Largely this is dependent on the stakeholders' application of the provided viewpoints and model kinds.

Changeability in remaining life-cycle

The changeability aspect of the production system applies to all the life-cycle stages of the production system, except perhaps the Exploratory research stage included in the life-cycle model from (INCOSE, 2011). Changeability in the remaining life-cycle most often takes the form of different structural divisions or changes in the system configuration encountered throughout the system life-cycle. PSAF is not intended to greatly describe these changeability aspects with dedicated models, but allow for generation of multiple models based on the same model kind, which can be used for describing the different structuring or configuration of the production system at various stages of the life-cycle, to address concerns relating to:

- Variations in structuring for…
  a) simulation purposes
  b) testing of partial systems by suppliers
  c) acceptance testing e.g. Site Acceptance test (SAT) and Final Acceptance Test (FAT)
  d) staggered equipment delivery, staggered run-in
- Reconfiguration for…
  a) service and maintenance accessibility
  b) continued operation in case of fault occurrence
  c) staggered shut-down or start-up

20.4 Assortment & hierarchy relations

It has been explained that the assortment & hierarchy phenomenon for production system architecture describes the recursive nature of production systems in regards to system hierarchies and system groupings. This means that production systems can be defined as belonging to the same families or assortments of production systems, and that systems exist on different levels e.g. that there is a system level hierarchy at play in the design of multiple production systems, so that a production system can consist of other production systems or can itself be a part of other production systems. The architecture descriptions derived from PSAF can describe this phenomenon, so that it is possible to not only identify the assortments of production systems and describe their common architecture; but also so that stakeholders may know the different uses of the different variants of production systems; and know-how and in what way the different production systems may be used in combination to design larger production systems e.g. in the way that machines are used in combination to form a production line. This use of architecture descriptions is particularly important for the management of production systems and among other for the purpose of defining production system platforms within the architecture; to plan and prioritize research and development; and to manage which production systems are used in the company's production and for what production purpose e.g. for what production locations, products or performance levels can a particular production system be used. The following main concerns have been identified in regards to the assortments and hierarchy context of production system:

a) Definition of assortment groupings (e.g. production system assortment/family/group etc.) for production systems, and the description of their combined architecture
b) The mapping/match between the production system assortment and the different production tasks of the company
c) The mapping between different levels of production systems and the possible configurations

21 Conclusion on a contribution to architecture descriptions

Years of research into the discipline of architecture and platform based design of products has shown that carrying out architecture-centric design and development means that the architecture is the focus of activities spanning almost the entire company organization. To successfully adopt an architecture-centric approach throughout so large a part of a company´s operation, means that we must not only be able to understand the architecture phenomenon, but also handle the architecture in an operational manner within a multidisciplinary environment involving both internal and external stakeholders. In order to do this it is essential to describe the architecture so that it may be addressed by multiple different stakeholders. Part 3 has contributed both an understanding of what architecture descriptions are and of a reference architecture framework (PSAF – Production System Architecture Framework), from which architecture descriptions can be generated for production systems. Such architecture descriptions describe key aspects of production system architecture based on stakeholder concerns relating to the system. The contribution to architecture descriptions has drawn on an international standard ISO/IEC/IEEE 42010 to formulate a conceptual model for architecture descriptions and reference architecture frameworks. This approach is hoped to provide a better basis for dissemination of the research results to industry professionals and other interested parties familiar with the ISO/IEC/IEEE 42010 standard or existing architecture frameworks in other sectors.

Part 4 and 5 of the dissertation will describe viewpoints, model kinds and correspondence kinds that constitute version 0.1 of this reference architecture framework. These viewpoints, model kinds and correspondence kinds provide support for production system designers in terms of production system architecture modeling and information handling.

Part 4 A contribution to viewpoints in a Production System Architecture Framework

Part 4 presents two viewpoints as part of a the suggested Production System Architecture Framework. The included viewpoints and modeling kinds are focused on description of the key constituent elements of production systems as well as the systems´ primary capability and changeability. The reference architecture framework can be used to govern descriptions of production system architecture to be used across stakeholder domains and system life-cycle stages, with a particular emphasis on the design phase of the system life-cycle.

The resulting architecture descriptions support decision making in the design process regarding the key design aspects of the production system to fit the system applications, which must be agreed upon by multiple stakeholders from different domains of the company i.e. different company functions, management levels, technical disciplines/backgrounds etc. The description also enables communication with external suppliers from early conceptualization on to later detailing, and can be especially helpful in communication before a detailed system specification is prepared or possible to make.

As part of the viewpoints, existing, modified and new models are used to describe different aspects of the architecture phenomenon. The model kinds are all described and their uses in description of different system architectures are explained. Examples and results from applying the reference architecture framework will be given from the primary case company.

22 Library viewpoints

There are many different ways to define the viewpoints of an architecture description. AGATE for example defined five viewpoints:

- View of Challenges, Objectives and elements of Context
- Business architecture view
- Service-oriented architecture view
- Logical architecture view
- Technical architecture view

To cover the selected concerns described in the previous sections, two library viewpoints have been defined within PSAF. The viewpoints consist of multiple model kinds, some that are already known to academia and industry, and a few new ones or modified versions of existing model kinds. The two viewpoints are:

Production capability Viewpoint (PCV): The Production Capability Viewpoint frames concerns relating to the production capability that should be achieved in the utilization stage of the production system life-cycle. As such the viewpoint frames concerns relating to the fulfillment of the specified production task.

Technical System Viewpoint (TSV): The Technical System Viewpoint frames concerns relating to the constituent design of the technical system, and how this design relates to the applications of the system. As such the viewpoint describes the constituent design of the technical system as framed by its composition, the relation to the desired capability, the changeability of the system, and the relations between the different constituent elements of the technical system. The viewpoint also frames concerns relating to the assortment and hierarchy relations for the system i.e. the connection to other production systems, showing how variants within an architecture for a family of production systems relate to each other, and how their relations may be determined by the applications in the company they influence or are influenced by.

To provide a full description of a production system architecture PSAF should include viewpoints that frame concerns relating to all the operators of the production system in order to properly describe the design layer of the architecture phenomenon. However this research project has focused on the design of the technical system and the specification of capability requirements for the technical process and the technical system. For this reason PSAF as it has been developed contains only the Production Capability Viewpoint, and the Technical System Viewpoint.

PSAF is intended to have a primary use in the design and procurement stages of the system life-cycle. As systems mature there is typically an expanding body of information and data available for the system, which could be incorporated in the architecture description. Because of the focus on the design and procurement of production systems for PSAF the included viewpoints are meant to give stakeholders an overview while also allowing for further detailed description of the architecture where needed (as judged by the stakeholders). The model kinds included in the two viewpoints support varying levels of detailing, and do not specify a specific level of detail. As such the models of the architecture description may be elaborated upon throughout the system life cycle.

The two viewpoints included in PSAF will be detailed in the following sections 23 and 24. Please note that most examples of the model kinds used in the descriptions are based on anonymized data to preserve confidentiality of the case companies. Minor formatting changes compared to the tested models have also been introduced. This has been done to aid in legibility; to consolidate modeling formalisms where alternative modeling has been tested; and to make it possible to include physically large models in the dissertation. It should also be mentioned that because architecture descriptions have such a wide and diverse field of applications, the descriptions do not fully describe all possible uses of the different viewpoints and their model kinds. Instead the descriptions provide examples of some of the most common uses and the relevant aspects of the modeling kinds.

23 Production Capability Viewpoint (PCV)

The Production Capability Viewpoint (PCV) addresses the concerns of system architects, product managers, production planners and other stakeholders involved in the specification and solving of a desired production task within the company. It is intended to be useful not only in the design stages of the system life-cycle, but also as a means of continually documenting the production capability as it evolves or changes throughout the system life-cycle. The PCV not only offers a snapshot of the system capability at initial production startup, but relates the capability to the planned or potential long term system evolution, which is subject to the requirements and constraints of many different system applications. As described in section 12, these applications not only cover the system’s role in the production function of the company, but also the system’s role in relation to strategies and operations of other areas of the company such as Product development, Finance, Human resources, Purchasing etc. In total the PCV allows for specification and communication of the core capability between stakeholders. Applying a capability view of the production system allows these stakeholders to answer questions such as:

- What is the range of input and output of the system?
- What is the processing capability of the system?
- What is the production capacity of the system?
- How is product variety generated?
- How are future product variants generated?
- What is the changeability of the system in relation to the production process?
- How does the system fit into the logistics chain?
- What new product or production technology will affect the system?

In general the PCV frames the core production task stemming from the requirements and constraints of different areas of the company, and it serves as the starting point for answering how the system design should realize the task. Specifically this view of the system capability is centered on descriptions of the range and capacity of the input/output of the system; the processing taking place within the system; the systems input/output logistics; and the changeability of all these. The PCV consists of the following model kinds that will be detailed further in the separate sections:

Table 3 - List of model kinds in the Production Capability Viewpoint

Model	Name	Description
PCV-1	Operand Master Plan	Provides a detailed description of the input and output range of operands processed by the system.
PCV-2	Process flow diagram	Describes the process flow within the production system covering different variants of outputs produced by the system.
PCV-3	Logistics diagram	Relates the production system to the logistics chain, and describes the logistics for operands to and from the system.
PCV-4	Production capacity plan	Describes the desired production capacity, forecasts production volume and documents the historical production volume.
PCV-5	Change impact roadmap	Describes the products and technologies that will or may affect the production system in the future.

Each of the model kinds included in the viewpoint are described in the following sections.

23.1 PCV-1 Operand Master Plan

The PCV-1, Operand Master Plan (OMP), provides a detailed description of what the production system is capable of producing and from what it is produced incl. any future production that the system must be prepared for.

What does it do?

The PCV-1 models the primary operands processed by the production system, while not including assisting or secondary inputs/outputs of the system. The operands are described by their composition and key attributes that can be objectively verified either by the production system itself or by other means. The PCV-1 also maps the link between the input and output of the production process and defines archetypes of output operands (types of operands sharing similar design characteristics), that can be used as collective references in other models and form the basis of focused system design. The PCV-1 models the operands in their input and output states.

What is it?

The PCV-1 is an object oriented modeling formalism based on the Product Family Master Plan (PFMP) modeling formalism (Harlou, 2006) with some modification. It consists of a number of tree structures describing the total range of operands; and a table that documents and maps the input and output states of specific operand variants and archetypes.

Intended usage

The intended usage of PCV-1 includes:

- Description and analysis of variety in system input and output to specify requirements for the system design
- Mapping between system input and system output
- Definition of archetypes of the system output
- Definition of future outputs or inputs the system design must be prepared for

Introduction to the modeling formalism

The PCV-1 provides an understanding of the production system's processing capability, as expressed through the systems capability to accept a given operand and process it to a desired output state. The processing capability only relates to the primary operands i.e. those operands whose processing is the very reason the production system exists. Secondary inputs/outputs (e.g. heat and noise) and assisting inputs (e.g. order information, cooling fluid and lubrication) are not covered by PCV-1. This does not mean that the PCV-1 could not be expanded to also cover these inputs/outputs; however this would require further development and testing of the modeling formalism.

To provide the intended overview the PCV-1 applies a production centered modeling viewpoint that describes the operands in their input and output states, and maps the two states to each other. This means that the operands are described through the characteristics that distinguish them from one another in the production system, rather than as viewed by other functions of the company e.g. product development, sales or purchasing. The PCV-1 also defines and describes archetypes of operands output from the system.

Type of model

The PCV-1 is an object-oriented modeling formalism based on the Product Family Master Plan (PFMP) modeling formalism that describes the variety of product families. The operands of a production system also constitute products or parts of products, and can be seen as a family of operands relative to that production system. It therefore makes sense to use the PFMP as a basis for describing the variety of the operands processed by the production system, even though the operands may not necessarily constitute finished products. The PFMP models product families as seen from a Customer, Engineering and Production perspective, by defining three views of the product family i.e. the Customer view, Engineering view and Part view respectively (not to be confused with the Views of an architecture description). Unlike the PFMP however, the PCV-1 describes a family of operands only from the production perspective and does so at different states (input state and output state). The PCV-1 thus applies the Part view to describe the operands of the production system, but does so twice to describe the operands in their input state and output state. This means that the PCV-1 contains two views on the operands i.e. the Input part view and the Output part view. Through these two views the PCV-1 describes the capability of the production system to handle a certain input of material, energy, information of living things, and to process these into the desired output. The two views can be described as follows:

Output part view: The output part view describes the variety of the operands that the production system can or should produce. The view describes the commonality and differences of the operands in their state when they exit the production system. The output should be described through use of attributes that can be objectively verified by the production system or once the operand exits the system. Both existing, planned or potential future output operands may be included in the Output part view to model planned or potential future capability.

Input part view: The input part view models the operands that are processed in the system. The view should describe the commonality and differences of the operands in their state when they enter the production system. The input should be described through the use of attributes that can either be verified by the production system or by other means prior to the operand entering the system. The input part view may include different operands that can be processed into the same output operand. This could be the case if the system allows for multiple processing options or if changes are made to the production system throughout the life-cycle e.g. addition or removal of equipment. Both existing, planned and potential future input operands may be included in the Input part view to model planned or potential future capability.

Modeling operands as systems

Using the PFMP modeling formalism as the basis of modeling the operands, means that the operands are in themselves considered to be systems. (Klir and Valach, 1967, p.21) defines a system as *"a set of interrelated elements"*, where an element is *"an indivisible unit whose structure we either cannot or do not want to resolve"* unless the resolution level is increased (Klir and Valach, 1967, p.35). The elements of the system are separated from the environment of the system by a boundary through which the elements of the system interacts with the environment (if the system is not absolutely closed)(see Figure 32) (Klir and Valach, 1967, p.28).

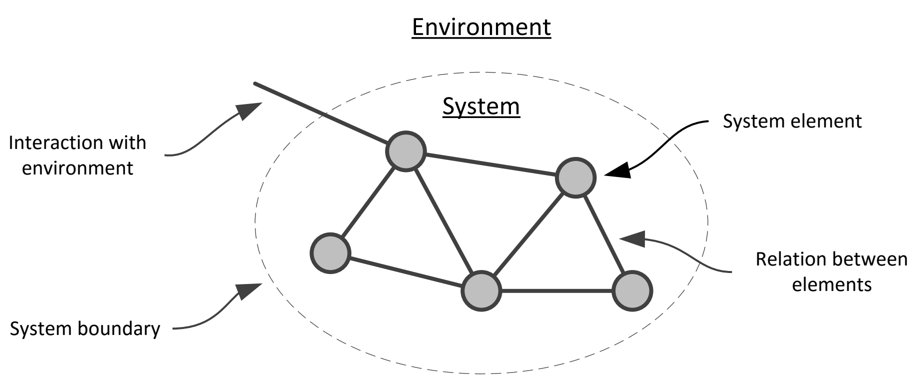


Figure 32 - Model of a system.

According to (Klir and Valach, 1967) a system is characterized by two basic properties: 1) its behavior, and 2) its structure. The PCV-1 models the operands perceived as systems, but only describes their constituent elements and structure, and not their behavior. The resolution level at which the operands are modeled is chosen by the architect, and is dependent on the sub-elements that are identified as being processed in the system. In some cases the operands may even consist of a single element, which is not sub-divided into smaller sub-elements e.g. a single workpiece is processed by the system.

Object-oriented modeling

The modeling formalism of PCV-1 falls within the domain of object-oriented modeling (OOM), as is the case for the PFMP modeling formalism which combines OOM with the Theory of Technical Systems and the Theory of Domains. OOM is part of the analysis and design activities within software engineering and computer science where an object-oriented paradigm is applied. That the modeling is object-oriented does not simply mean that the model concerns itself with the modeling of individual elements, but entails that it is based on the concepts of the object-oriented paradigm used in object oriented design (c.f. section 5.5). The PCV-2 applies an object-oriented modeling of operands by modeling the class hierarchies of operands.

Different relationships between objects define the hierarchies within the object-oriented paradigm. The PFMP modeling formalism models three types of such relationships. These are the Generalization-specialization relationship, Whole-part relationship and Instance relationship. The object-oriented paradigm distinguishes between strong and weak whole-part relationships, known as Composition and Aggregation respectively. The definition of the PFMP modeling formalism (Harlou, 2006) does not distinguish between these two, and also refers to the Whole-part relationships as Aggregation, despite examples and descriptions in the formalism clearly demonstrating only Composition relationships. In the modeling of products or operands processed by a production system, this distinction is important when sub-elements can be a part of many different assemblies e.g. a sensor that has the option of being built into two different modules. The PCV-1 corrects this shortcoming of the PFMP and thus models four types of relationships Generalization-Specialization, Composition, Aggregation and Instance (see Figure 33).

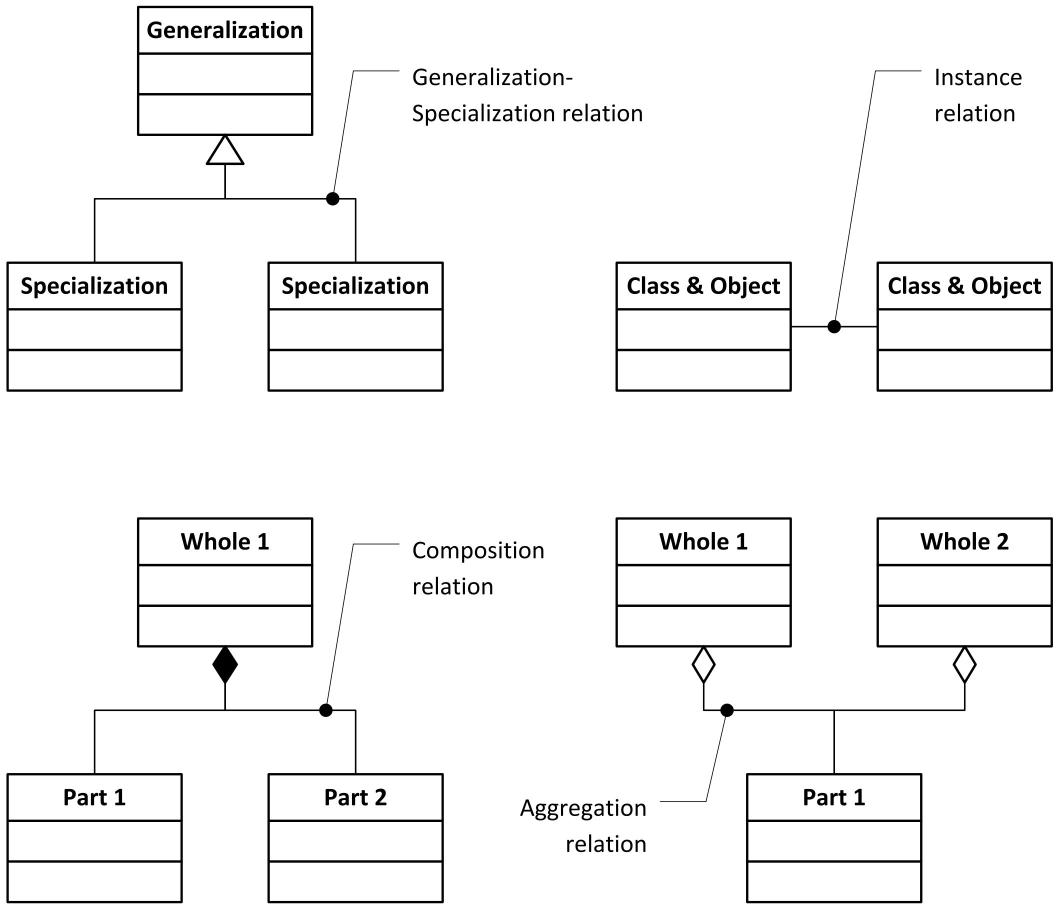


Figure 33 – Select examples of object & class relationships in the object-oriented paradigm.

The four relationships can be described as follows.

<u>Generalization-specialization relationship:</u> The generalization relationship describes a parent/child or kind-of relationship between elements, where one element is a specialization of another element and inheritance of structure and/or behavior occurs from the parent to the child. For example a Mountainbike is a specialization of a Bicycle, and conversely a Bicycle is a Generalization of a Mountainbike. Generalizations of elements are referred to as super-types or super-kinds and specializations of elements are referred to as sub-types or sub-kinds. Modeling of the generalization relationship for operands is used to describe the variety in types of operands, with sub-kinds inheriting attributes and constraints from their super-kinds.

<u>Composition relationship:</u> The composition relationship is a type of whole-part relationship. The composition relationship describes a part-of relationship between elements, where one element is a part of a whole e.g. the way that the class Bicycle consists of the class Wheel (among other). The constituent elements of the whole are called sub-parts, and the whole of which they are a part are called super-parts. The composition relationship represents a strong association between elements, where the destruction of the super-part normally will also entail the destruction of any sub-part having a composition relation to it.

Aggregation relationship: The aggregation relationship is a type of whole-part relationship. The aggregation relationship describes a part-of relationship between elements where an element is a part of a whole. Unlike compositional relations however an element may have an aggregation relation to multiple other elements, meaning that an element may be a part of multiple wholes. The aggregation relationship represents a weak association between elements, meaning that if the containing element (aggregate) is destroyed, then the contained element is not necessarily destroyed as well. Similar to the composition relationship the containing elements are referred to as super-parts and the contained elements with aggregation relations to the aggregate are referred to as sub-parts.

Instance relationship: An instance relationship is *"a model of problem domain mapping(s) that one object needs with other objects, in order to fulfill its responsibilities"* (Coad and Yourdon, 1991, p.127). Instance relationships represent a dependency between elements. In the case of modeling operands, this instance relationship exists both in the configuration constraints of operands and in the dependency between the input and output operands, where production of particular operands are dependent on operands in specific input states. It should be noted that a particular output operand may have multiple instance connections to different input operands, due to the fact that the output could potentially be generated from multiple different inputs depending on the processing options of the production system.

What the PCV-1 models

The PCV-1 models operands in their input and output states, or to put it another way, it models the operands entering and leaving the production system. The model only models the static aspects of the operands and not the dynamic i.e. behavioral aspects. This is because the model does not concern itself with the behavior of the operands either in the production or any other life-cycle stage. The PCV-1 adds to the PFMP modeling formalism by also modeling the objects/instances of operands and the aggregation relationship. The model consists of three main elements (see Figure 34):

- Part-of structure
- Kind-of structure
- Variant table

The Part-of and Kind-of structures describe the range of input and output operands that specify the total output capability of the production system, and the variant table provides a view of the specific input and output that the system must or could potentially produce.

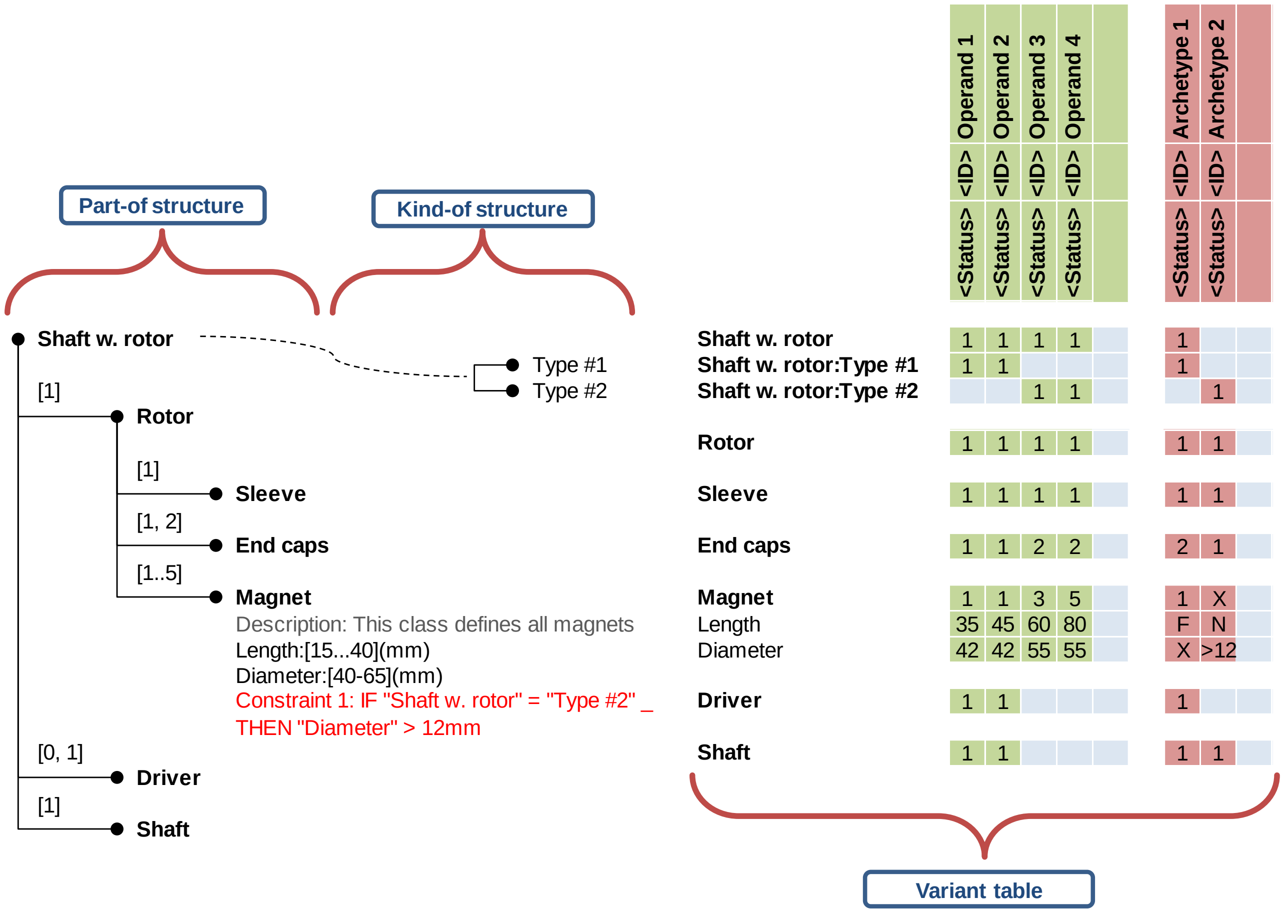


Figure 34 - The three main elements of the Operand Master Plan (PCV-1)

Class definition

The Part-of and Kind-of structures follow the modeling notation used in the PFMP, with a few minor adjustments. The structures contain classes of operands along with their associated attributes and constrains**.** The part-of structure models the composition and aggregation relationships between classes of operands, and the kind-of structure models the generalization- specialization relationship between classes of operands.

The classes of operands in PCV-1 are defined by the architecting stakeholder. Often these classes can be based on documentation of the workpiece to be produced e.g. Bill-of-materials, workpiece drawings etc. The classes encapsulate a description of one or more operands sharing characteristics. Each class is given a unique name and optionally a description if the name of the class is insufficient to communicate what operand the class name relates to. The definition of classes represents an abstraction of the actual operands whereby the operands are described only through a select number of aspects e.g. material, dimensions etc.

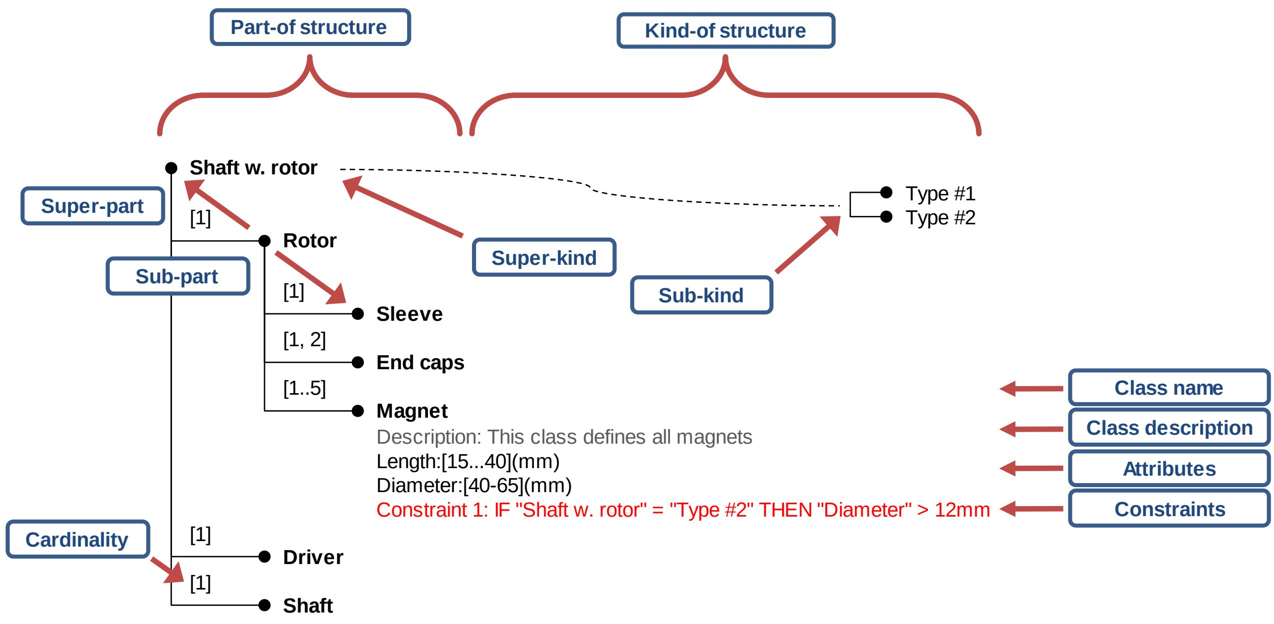


Figure 35 - Part-of & kind-of structures

Part-of structure

The Part-of structure models composition and aggregation relations as a tree structure. Within the Part-of structure super-classes and sub-classes are referred to as super-parts and sub-parts (see Figure 35). Each class in the Part-of structure is represented by the class name and a node in the tree structure. Sub-parts that have a composition relation to its super-part are represented by a filled circle/node in the tree-structure. Sub-parts that have an aggregation relation to their super-part are represented by an unfilled circle/node in the tree-structure. Note that sub-parts with aggregation relations will appear multiple places in the tree-structure.

Classes in the part-of structure are given cardinality which indicates the number of sub-classes that are a part-of a super-class e.g. a bicycle has two wheels whereby the cardinality of the class “Wheels” is [2]. Cardinality is expressed as a number, a set of numbers or a range e.g. [1], [1, 2, 4] or [1..9] respectively. In the case of the Input part view it is often sufficient to describe a single level part-of structure.

Kind-of structure

The Kind-of structure models specialization/generalization relations. Within the kind-of structure classes are referred to as super-kinds and sub-kinds (see Figure 35). Sub-kinds are placed in a single level structure and connected by a dotted line to their super-kind to form the kind-of structure. Sub-kinds may in themselves be super-kinds for other sub-kinds, meaning that the modeled sub-division of kinds of operands may be multilayered e.g. a Cross country mountain bike is a sub-kind of Mountain bike, which is itself a sub-kind of Bicycle.

The classes in both the Part-of and Kind-of structure can have attributes that describe variety within the class and constraints that describe the possible combinations of classes (see Figure 35).

Attributes

Attributes describe the variety within a class of operands. These attributes should be verifiable either by the production system, or when the operand enters or exits the production system. (Coad and Yourdon, 1991, p.119) defines an attribute as *"some data (state information) for which each Object in a Class has its own value"*. PCV-2 uses the following types of attributes:

- *Identifier:* Expressed by a text string e.g. [Ferrite, NdFeB]. In the PCV-2 one or more letters of the string are highlighted by use of bold and/or underline e.g. [**F**errite, **N**dFeB]. The highlighted letter(s) are used in the cells of the variant table instead of the full text string (since this may be too large for the cells of the table).
- *Integer:* Are whole numbers and may be either positive or negative. The attribute may be expressed as either a range of number e.g. [-1…3] or a set of numbers e.g. [-1, 2, 3]
- *Real:* Real numbers are rational or irrational numbers. The attribute may be expressed as either a number range e.g. [-3.5…5.7] or as a set of numbers e.g. [-3.5, 1.7, 5.7]
- *Boolean:* Is an attribute which may be either true or false e.g. Open end [True, False]. In the variant table Boolean attributes are referenced by their starting letter e.g. T or F (since the full text may be too large for the cells of the table)

Sub-kinds inherit attributes from super-kinds. Attributes that are shared by more than one sub-kind is placed only in the super-kind to which they are related.

Constraints

Constraints on combinations of operands can be defined using different formats. These constraints define how the operands may be combined e.g. what input operands are used together, and which are not e.g. either magnet powder is the input (which is then compressed to a workpiece by the system), or a pre-compressed magnet is the input (whereby there is no compression sub-process carried out by the system). PCV-2 uses the following types of constraints:

- *Verbal:* Verbal constraints are expressed by sentences e.g. "Magnet powder and Pre-magnets are not used together"
- *Logic:* Constraints expressed by logic e.g. Magnet powder -> NOT Pre-magnet
- *Calculation:* Calculations are mathematical constraints e.g. Rotor_weight = Magnet_weight + Shaft_weight
- *Combination table:* Combination tables map the number of the different operands used in combination. The location of combination tables is not specified, but it must not interfere with the variant table. The variant table in itself maps some of the actual or possible combinations, but does not fully describe constraints on combinations such as a combination table does. The variant table describes instances of the input/output variants.

Variant table

The variant table (see Figure 37) models two things 1) variants of individual operands that are output from the system, as well input operands which are used in their production 2) and archetypes of output operands. The variants of operands may both be ones already existing or variants that the system must potentially be capable of processing in the future. The variants of operands constitute instances of the operand classes modeled in the Part-of and Kind-of structures. These instances are described through the cardinality of the classes from which they are derived and the values of the attributes of these classes. In this way the variant table shows the composition and characteristics of specific output operands and the input operands from which they are produced.

Archetypes of output operands represent the typical designs of operands which the system must be capable of producing. Defining archetypes aid in both communication and referencing of output operands in other contexts. The definition of archetypes is subjective and is determined by the system architect. It may be based on a similarity of attributes or it may be based in a market perspective, logistics perspective, product perspective etc. The determining factor ought to be a combination of a processing and part perspective, in that parts with similar processing or attributes are determined to be of the same archetype i.e. the same design type. These definitions of archetypes may be used across architecture descriptions, and can constitute a common way of describing the types of operands used within the company's products. Only the variety in the Output part view is documented for archetypes because an output can be processed from many different inputs.

Columns of the variant table

Each column in the table represents a variant of an output operand or archetype (see Figure 36). Modeling in both the Input part view and output part view makes it possible to see which input operands are processed into a given output. The same output operand may appear in more than one column if the production system can produce the output from different sets of input operands e.g. the input could be either raw sheet metal or a pre-processed component. The variety in both the Output part view and Input part view is documented for output operands, whereas the variety in the Input part view does not necessarily have to be documented, unless a specific pairing of input and output can be determined for the archetypes.

	<Status> <ID> Operand 1	<Status> <ID> Operand 2	<Status> <ID> Operand 3	<Status> <ID> Operand 4		<Status> <ID> Archetype 1	<Status> <ID> Archetype 2		
Super-part A	1	1	1	1		1	1		← Cardinality
Attribute 1	F	F	N	N		F	N		← Attribute value
Attribute 2	35	50	80	90		<55	<95		
Super-part A:Sub-kind 1	1	1				1			
Super-part A:Sub-kind 2			1				X		
Super-part A:Sub-kind 3				2			X		

Figure 36 – PCV-1 variant table format

The following is given for both specific output variants and archetypes:

- **Name:** Name of the output operand or archetype
- **ID:** Any identification used within the company e.g. part number or documentation reference such as a drawing number.
- **Status:** The current status. This may be highly company dependent, but could for example specify if the operand is currently in production within the company, under development, or merely a potential variant to be produced in future.

Rows of the variant table

Each row in the table represents one class or attribute (see Figure 36). The characteristics of both specific variants and archetypes are described by the cardinality of their classes and values of their class attributes. The table would for example document that a specific output operand variant #5551 contains 1 Ring magnet, and that the magnet is made from Strontium Ferrite. It would also show that the output variant is produced from (among other) 52 grams of Strontium ferrite powder (see Figure 37).

Classes are referenced by their name and, if necessary, the names of their super-classes. The sub-class "Ring magnet" could for example be referenced as either:

Sintered ring magnet <u>or</u> **Rotor.Magnet:Ring magnet**

If super-classes are included in the reference for the class, then class names are separated by periods for super-parts and colons for super kinds, but users are free to substitute the naming syntax with one of their own choosing. Attributes are merely referenced by their names.

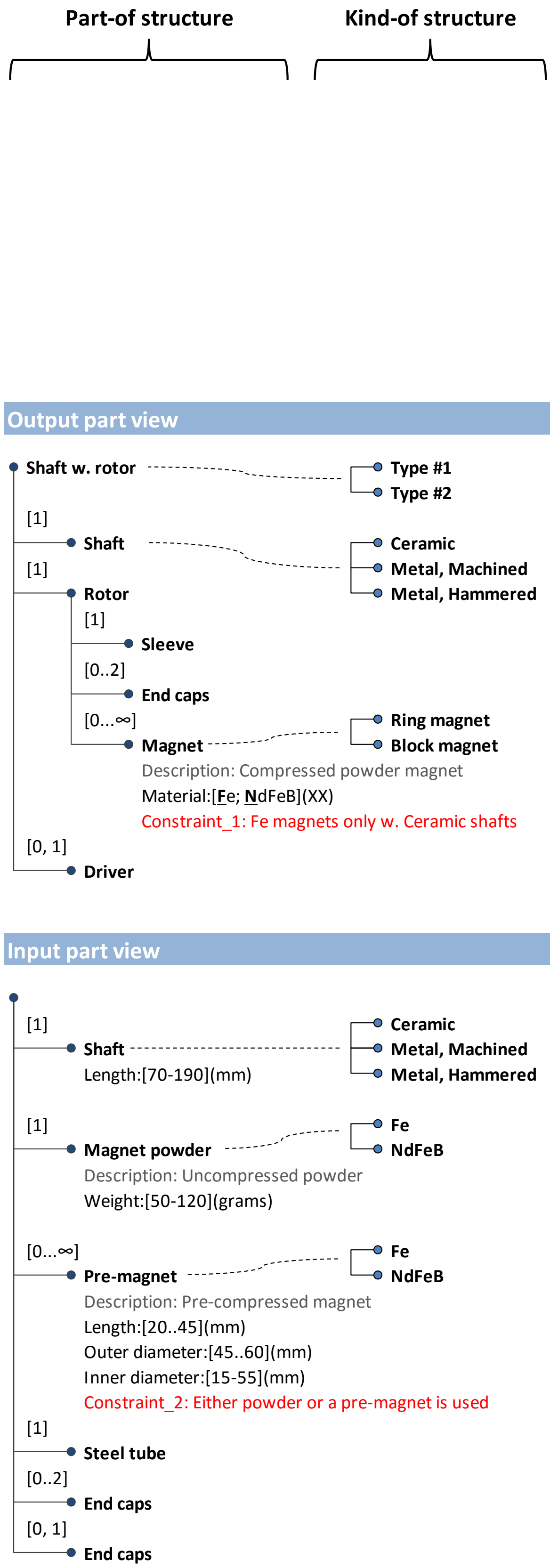
Cell entries in the variant table

The cells of the table document the values of class cardinality and attribute values for each of the system outputs. As a general rule, attributes and cardinalities follow the same syntax as in the Part-of and Kind-of structures, but the following should be noted when mapping:

- Attribute values for archetypes may be a range or set.
- In cases where an archetype may be composed of different sub-kind, the cardinality of each class of sub-kind can be given as "X".
- Attribute values of the Identifier or Boolean types can be referenced by one or more representative letters. The representative letters are written underlined and/or bold in the full attribute name under the class. For example:

Attribute:	Color: [<u>**R**</u>ed, <u>**G**</u>reen, <u>**B**</u>lue](String)
Letters used in variant table:	R, G, B

Figure 37 shows a reduced and censored example of a PCV-1 from Grundfos. The production system in question produces magnetic rotor assemblies (Shaft w. rotor) to be used in electric motors. Please note that the Sub-kinds of the super-part "Shaft w. rotor" correspond to the defined archetypes.



Variant table

Output part view

	Produced #5551 Variant 1	Produced #5552 Variant 2	Development #7771 Variant 3	Potential #7772 Variant 4			Produced A1 Archetype 1	Development A2 Archetype 2
Shaft w. rotor	1	1	1	1			1	1
Shaft w. rotor:Type #1	1	1					1	
Shaft w. rotor:Type #2			1	1				1
Shaft	1	1	1	1			1	1
Shaft.Ceramic	1	1					1	
Shaft.Metal, Machined			1					X
Shaft.Metal, Hammered				1				X
Rotor	1	1	1	1			1	1
Rotor.Sleeve	1	1	1	1			1	1
Rotor.End caps	2	2	2	2			2	2
Rotor.Magnet	1	1	2	5			1	X
Material	F	F	N	N			F	N
Rotor.Magnet:Ring magnet	1	1	2	3			1	X
Rotor.Magnet:Block magnet								
Driver	1	1					1	

Input part view

	Variant 1	Variant 2	Variant 3	Variant 4		
Shaft	1	1	1	1		
<Length>	72	72	182	182		
Shaft.Ceramic	1	1				
Shaft.Metal, Machined			1			
Shaft.Metal, Hammered				1		
Magnet powder	1		1	1		
<Weight>	52		75	75		
Magnet powder:Fe	1					
Magnet powder:NdFeB			1	1		
Pre-magnet		1	1	1		
Weight		55	75	75		
Pre-magnet:Fe		1				
Pre-magnet:NdFeB			2	3		
Steel tube	1	1	1	1		
End caps	2	2	2	2		
Driver	1	1				

Figure 37 - Example of PCV-2 for a magnetic rotor sub-assembly in an electric motor.

Remarks on development

The PCV-1 has been tested in development projects for production systems only with modeling of Material operands (materials, components, assemblies). Other operands such as Energy, Information and Living things were not included in the model, as these were not processed by the production systems in question. Further testing will be necessary to determine it the model is as effective in modeling of other kinds of operands, though no major problems are expected.

Testing of the modeling formalism on a family of production systems showed that it provided stakeholders with a good overview and documentation of the input and output processing capability of a family of production systems. However, it was also seen that for production systems or families of production systems producing many different variants the variant table can become quite large, especially when printed. This can be remedied by either choosing not to print the full model, or by selecting only key representative outputs to be shown in printed versions. Viewing the model in a full scale allows for visual identification of groups of output variants with similar characteristics. Representative outputs can be chosen for these groups, and archetypes can also be determined. Conversely it is also possible that the PCV-1 becomes very small, in the case of production systems that are not intended to produce a great variety of output operands e.g. only one variant. This is particularly true when the architecture covers only a single production system, and when the system is at the level of a single machine or lower. In these cases stakeholders may find that the model kind is needlessly complex. The PCV-1 is concluded to be most useful for description operand variance when this variance is substantial.

23.2 PCV-2 Process flow diagram

The PCV-2, Process Flow Diagram, describes the process flow(s) of the production system for different operand archetypes or specific operand variants. The model kind provides a consolidated view of the production processes carried out by the production system, and the variations in flow experienced by different operands.

What does it do?

The PCV-2 provides a description of the production process (technical process) with the different possible flow variants necessary to deliver the variety in the production system's output. The model kind provides a static view of the process flows, and does not describe dynamic changes in flow within the system. Dynamic models of the production system are not included in the current version of PSAF, but could be included in future, either as a part of the Production Capability Viewpoint or a separate viewpoint addressing stakeholder concerns regarding the dynamic behavior of the production system.

What is it?

The PCV-3 Is a structural model of the production process showing the sub-processes of the production and the different possible flows between sub-processes. A notation is suggested for the modeling, but PCV-2 does not dictate the use of any one particular process modeling notation. The only requirement for the applied modeling notation is that it be capable of showing the process flow through the system and any differences in flows for the various operand archetypes or specific variants.

Intended usage

The intended usage of PCV-2 includes:

- Analysis and synthesis of the production process on a sub-process level.
- Main reference for the production process in the production system.

Introduction to the modeling formalism

The PCV-2 models the processing of operands in the production system. The focus is on the primary operands, whereas processing of secondary or assisting inputs is not included e.g. the filtering of lubricant for a process. The model includes information on the sub-processes and the variety in flow between sub-processes.

Type of model

The PCV-2 is a structural model that may be based on different process flow modeling notations. Notations from UML, BPMN etc. may all be used, as long as they model sub-processes, variations in flow paths and the input operands. Whatever the modeling notation chosen, it is a requirement that the sub-processes modeled in the process flow diagram(s) are given names and/or other identifiers such as a RDS ID (see Part 5).

The PCV-2 models sub-processes and the flow between them for different operands. Alternative flow paths are included both to distinguish between different variants and to model multiple processing options for the same operands. The process flow should be supplemented with a table listing the relevant process numbers for the operand archetypes, or it should be indicated in the diagram which archetypes or specific operand variants follow which particular flow paths.

Example

Figure 39 shows an example of the PCV-1 for a production system assembling a rotor for a circulation pump. The process includes processes associated with processing of input operands belonging to existing rotor designs to be produced. The example follows the notation shown in Figure 38

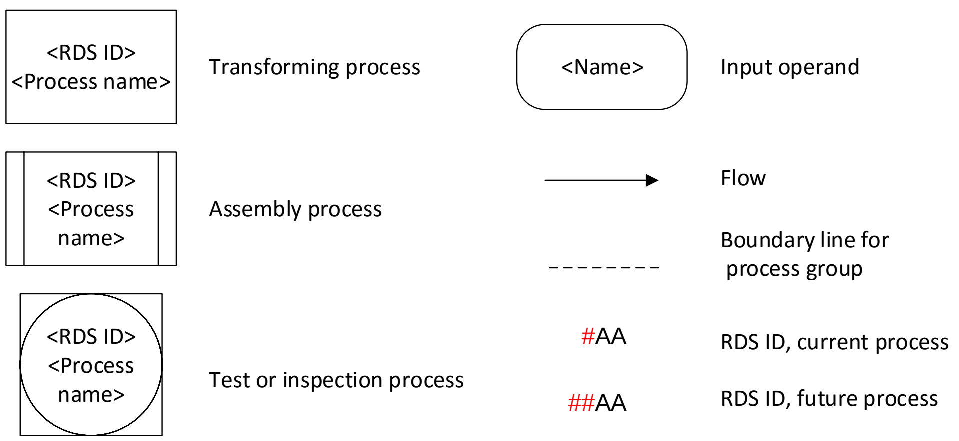


Figure 38 - Legend for PCV-2 process flow diagrams.

If non-existing sub-processes e.g. future processes are included in the flow diagrams, it is advised to indicate their status by some unique means e.g. by giving the process shapes a unique color or pattern. In the example in Figure 39 two future processes and three future Input operands are included in the process flow and given a hashed pattern to indicate that they are future or potential Inputs and sub-processes. These may be added at a later time in the production system life-cycle, at which time they may replace some of the other inputs and sub-processes.

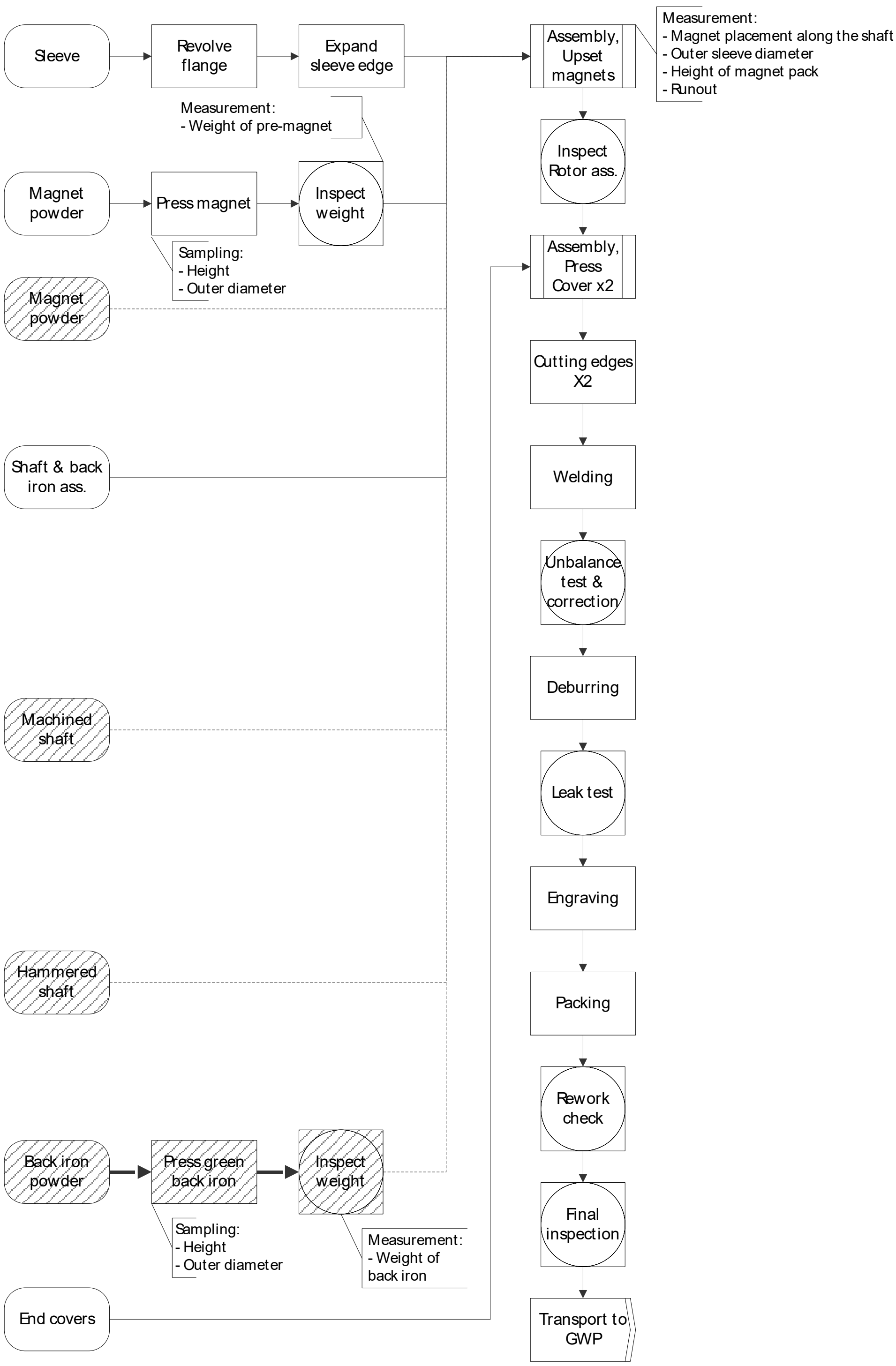


Figure 39 - Process flow for a rotor assembly line.

Remarks on development

The PCV-2 represents a simple process flow diagram, which is a model kind that is already in widespread use in industry. As such testing of this model kind has not been the subject of much separate testing. The use of the RDS ID as references for the different sub-processes was however tested separately in the development project for the assembly line represented by the process flow shown in the example. The RDS ID's were shown to provide a good point of reference in communication with equipment suppliers for the line, and as a design reference for the development team. Among other the sub-processes and ID's were used in structuring specifications for the production line, and used to provide a link between process and equipment designs.

23.3 PCV-3 Logistics diagram

The PCV-3, Logistics diagram, describes the production system's role in the logistics chain of the production it is intended for. This description constitutes part of the specification for what role the production system is capable of fulfilling. The PCV-3 also addresses concerns relating to the physical/practical logistics surrounding the input and output to and from the production system i.e. how are inputs presented to the system and how do processed operands leave the system.

What does it do?

The PCV-3 describes the immediate logistics chain of which the production system is a part. It further more describes the state of input operands as they are delivered to the system and of output operands as they leave the system. This includes the means of transport to and from the system, the physical state of the input and output, and the means by which it is transported to and from the system.

What is it?

The PCV-3 is a simplified visual representation of the logistics chain, showing locations or production systems and the logistics between these. The model kind also describes the means by which the operands are transported and their state at input and output. This overview can aid in the specification of other systems, and it serves as a reference for requirements in handling of inputs and outputs i.e. it specifies how the input is presented to the system, and it specifies how the output should be delivered from the system in terms of the operands state and the means of transport (is the output for example packaged, or simply placed in a particular location.

Intended usage

The intended usage of PCV-3 includes:

- Give a basic specification of how the output from the production system should leave the system (state, means, amount)
- Give a basic specification of how the input operands enter the system (state, means, amount)
- Provide an understanding of the production systems role in the logistics chain. The PCV-3 describes the relation between the production system and other production systems or other company entities in terms of their deliveries to one another.

- Provides and understanding of the production systems application in the production function of the company and provides an understanding of any requirements and constraints on the system design that may arise from the systems application in the production function of the company.

Introduction to the modeling formalism

The aim of PCV-3 is to provide a view of the logistics flow in the immediate environment of the production system (of relevance to the production system). The focus in this description is on the primary operands of the production system, secondary operands and assisting inputs are not covered by the model kind, apart for assisting inputs involved in the transport of operands e.g. packaging. The model shows the state of the operands as they enter and leave the system, and the means by which the operands are delivered to and from the system e.g. packaging

Type of model

The PCV-3 is a structural and pictoral model that consists of two diagrams 1) one showing the logistics flow, and 2) one describing the state of the operands and the means of the transport to and from the production system i.e. the:

- Logistics flow diagram
- Transport means and state diagram

Logistics flow diagram

The logistics flow diagram is a visual modeling formalism based on the simple notation shown in Figure 40. The diagram depicts the transport between production locations or production systems of either primary operands (specific operand variants, archetypes or groups of operands); or assisting inputs/outputs involved in the logistics to and from the system (e.g. packaging).

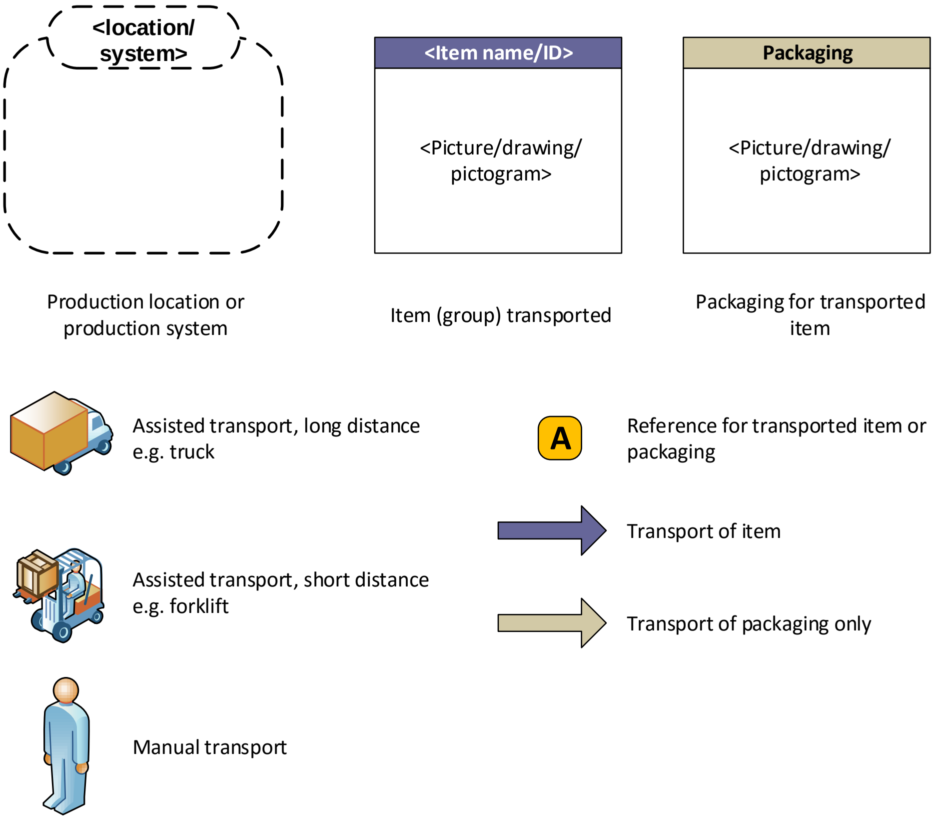


Figure 40 - Notation for logistics flow diagram.

The elements of the diagram can be combined as shown in Figure 41.

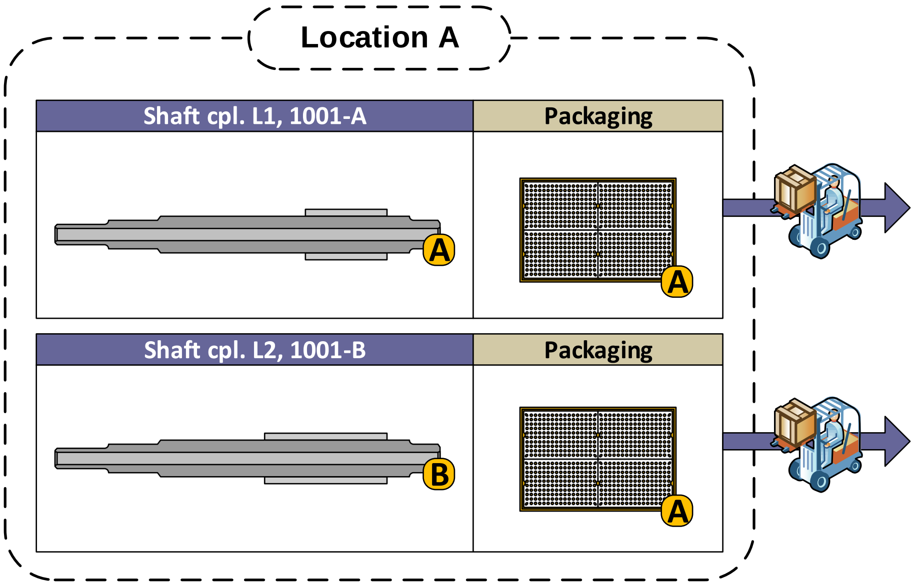


Figure 41 - Example of logistics from a production location. Two variants are shipped using the same packaging.

Transport means and state diagram

The transport means and state diagram describes the means (assisting inputs/outputs) involved in the transport to and from the system, as well as the state of the transported items. The diagram is based on the modeling notation shown in Figure 42. Each means of transport has an individual box where it is described. The state of the transported item is also described for each means of transport. The following information is included in the description.

- Symbol for the means/packaging
- Name of the means/packaging
- Pictorial diagram describing the means/packaging
- Physical state of the transported item
  a) Relevant physical attributes of the means e.g. dimensions, weight, material
  b) Capacity of the means for a particular item
  c) Reference for the transported item e.g. drawing or symbol
  d) Description of the state of the operand in relation to the packaging (how is the item packed or ordered). This can be shown with a drawing.

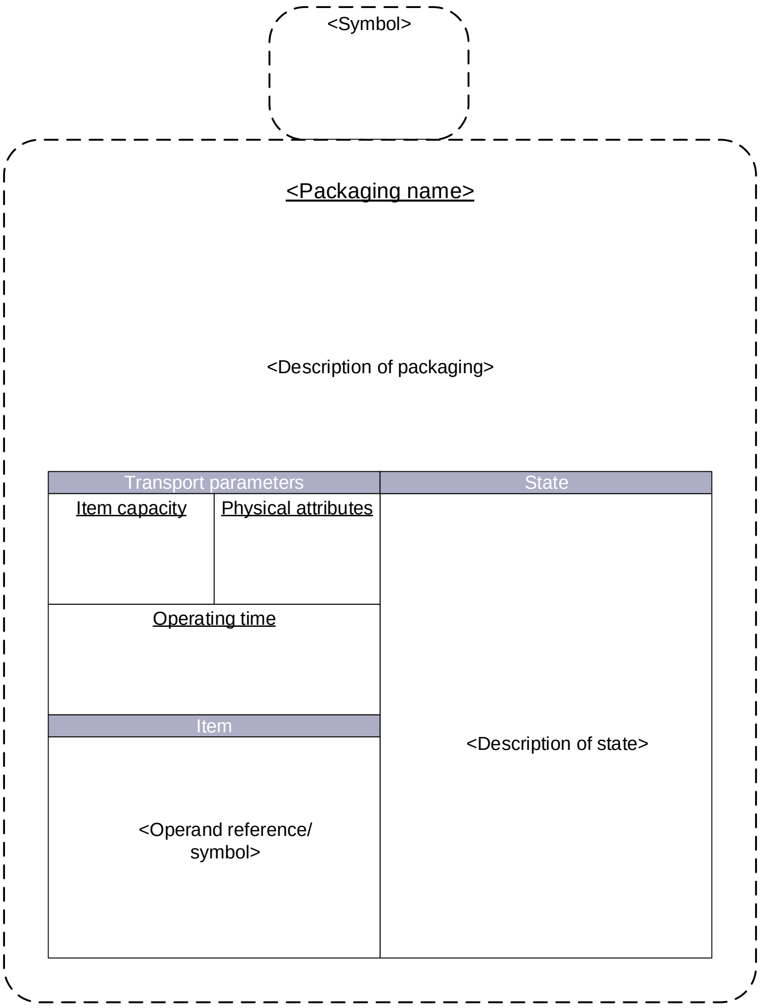


Figure 42 - Description boxes for supplies?

An example of one description of a transport means is shown in Figure 43. The example depicts a pallet and plate combination used as the means of transport for six different variants of operands input to and output from the production system in question.

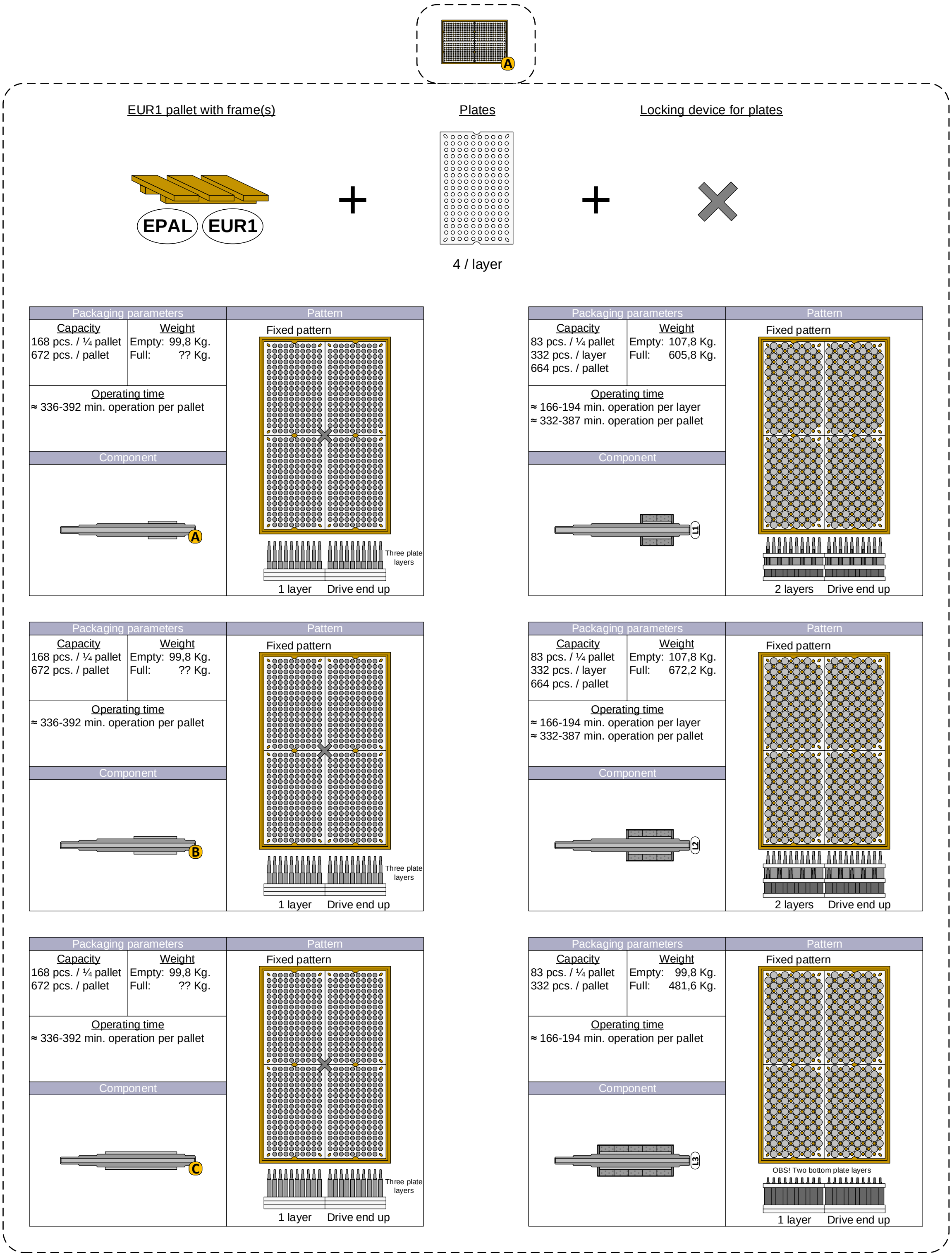


Figure 43 - Example of a description for a means of transport. A pallet used for transporting 6 different operand variants is shown.

Example

Figure 44 shows an example of the PCV-3 for a production system producing rotors for circulation pumps. The logistics flow diagram contains two versions: One that models the logistics immediately after the initial start of production, and one showing the logistics flow a few years after start of production, at which point changes are made to the total logistics involved in the production of the circulation pumps. The rotor to be produced is replaced with a new planned design, and changes are made to the logistics situation i.e. the production system now needs to accept a different input and output a new rotor variant. Changes to the sub-processes involved with the production can be seen in PCV-2.

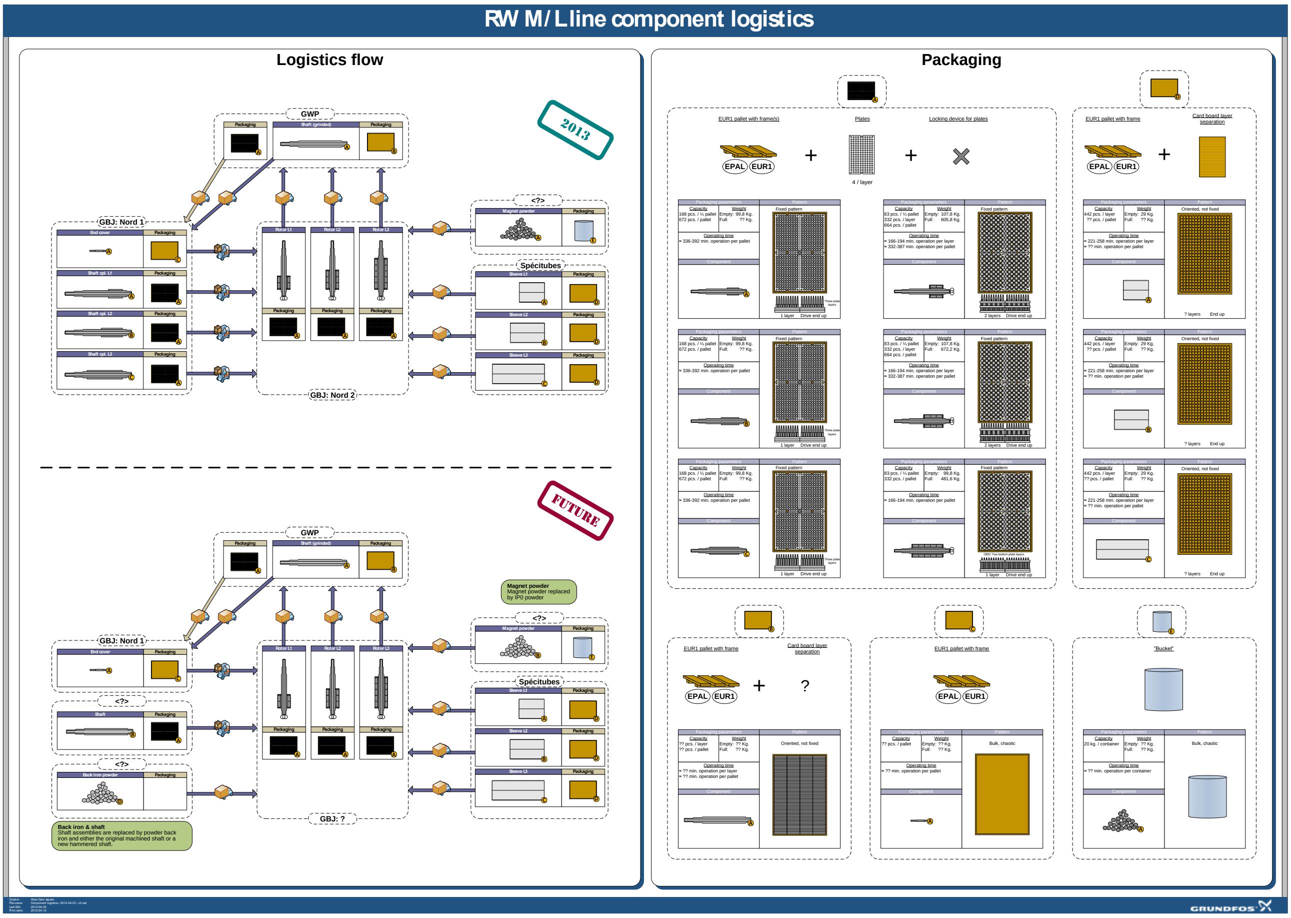


Figure 44 - Example of PCV-5 for a production line at initial start of production and future

The output of the system could either be a place where the operand is presented in a packaging, or it could be an interface to other production system.

Remarks on development

Development and testing of the PCV-3 has been conducted as part of the procurement of new production equipment. The model kind has demonstrated an advantage in providing a clear overview of the logistics context of a system to be procured, which is otherwise described by text alone. The description of the state of the operands as they leave or enter the system is particularly relevant for the specification of the systems handling capability. Additionally the PCV-3 provides an understanding of the system's role in the production of a product or family of products by way of its role in the total production of a product. The PCV-3 is concluded to be mostly useful in the description of architecture for single production systems, because the logistics context of systems within a production system family can wary greatly.

23.4 PCV-4 Production capacity plan

The PCV-4, Production capacity plan, provides stakeholders with an overview of the projected, planned or historical production capacity and production volume of the system. The PCV-4 also describes the financial investments in the production system throughout the life-cycle, and links it to the changes in production capacity. The overview serves as both a specification of capacity and investment, and also serves as the means of documenting capacity, production volume and investments in relation to the production system.

What does it do?

The PCV-4 describes required or projected production capacity for the production system, and documents projected and historical production volumes divided by output variants, groups of variants or archetypes. The PCV-4 also describes the realized or planned/projected investments for the production system. Capacities, production volumes, and investments are given by year (or other increment depending on the resolution level required by the stakeholders). The PCV-4 in its current form covers only production systems whose input and/or output is either material or living things. Production systems that process information or energy are not covered by the modeling formalism.

What is it?

The PCV-4 consists of a simple table and graph showing the capacity and production volume that constitutes the requirements for the system's capacity. Also included are the investments that are planned or already incurred throughout the system life-cycle, as a result of the initial system build and subsequent change.

Intended usage

The intended usage of PCV-3 includes:

- Provide a precise specification of the system's required minimum capacity and rated capacity throughout the system life-cycle
- Document historical or projected production volumes, capacities and investments
- Show the link between investment and changes in capacity and production mix

Realized or projected production volumes are given either for specific output variants, groups of variants or archetypes. Along with separate estimates of the corresponding cycle times of each output this allows for specification of the minimum required production capacity of the production system throughout the life-cycle. Appropriate changes in the rated capacity of the system and the necessary accompanying investments can then be similarly specified throughout the life-cycle. Specification of the changes in rated capacity and investments must follow consultation with relevant stakeholders and analysis of requirements and constraints of the system architecture.

Because the capacity requirements and mix of variants produced by the system will be dependent on other production systems, it can often be a good idea to consolidate the PCV-4 of different architecture descriptions to balance the capacities and production mix of the systems. The PCV-4 thus also serves as the input for other the design of architectures of other systems.

Introduction to the modeling formalism

The PCV-4, Production capacity plan, provides an overview of the requirements for the production system's production capacity and investments. The production capacity is specified in appropriate measurements and time intervals e.g. produced units/weight/length per year/month/week. The PCV-4 is also used to document historical production data and basic information regarding the location of the production system. The model can be used as a means of documentation, and a means of discussions of concerns relating to the capacity, production mix and investments.

Data/information in model

The PCV-4 contains data based on both documented production data as well as projections and plans. The following data is included:

- Identifying information for the production system
- Production system locations
- Historical data, projections or plans for:
  a) Production volume
  b) Rated peak production system capacity
  c) Investments
- Production mix incl. start and end dates of production and reference to the intended product

The production mix i.e. the mix of output variants produced by the system, can be divided either by individual variants, groups of output variants (e.g. variants intended for a particular product family), archetype (as defined in PCV-1) or any other appropriate grouping (e.g. grouped according to the source of estimates for example sales companies or markets).

It should be noted that the rated peak system capacity is dependent on a number of factors including but not limited to:

- Available production time pr. year
- Production performance e.g. Overall Equipment Effectiveness (OEE)
- Cycle time per produced unit
- Production mix (mix of output variants produced)
- Production strategy type
  a) Make to stock
  b) Make to order

Changes may occur to these factors with varying degree of impact on the rated capacity. Whenever major changes occur to the different factors, the PCV-4 should be updated and consulted to re-asses the capability requirements of the system.

Type of model

The PCV-4 is primarily a tabular model supplemented by a graph depicting some of the key data along a time axis.

Structure of the model

There are two elements to the PCV-4, a table and a graph showing parts of the table data in graphical form (see Figure 45). If the architecture description describe the architecture of multiple systems, then the tables and graphs may be used to cover groups of systems within the architecture.

Tables in the model kind includes:

- Identifying information for the production system
- Information regarding where the system is located
- Information on the produced output (e.g. archetypes or specific output variants)
- Yearly information on production volumes.
    a) Previous years show historical production volumes
    b) Future and current years show projected or required production volumes
    c) Investment in the system by year
    d) Estimated or calculated maximum capacity

A finer resolution of the production data can be chosen if it is relevant, it does not have to be on a yearly basis. The production data can be shown in the graph in PCV-4. At a minimum the accompanying graph should show:

- Investments by year
- Total investments
- Rated peak system capacity (given maximum possible OEE and utilization)
- Total production volume

System info		Output info		Production dates		Past			Present & future				
System name	**System ID**	**Output variant**	**Intended product**	**Start**	**End**	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**
<Name>	<ID>	<Output variant>	<Product name>	<Date>	<Date>								
Location	**Transfer date**												
<Location ref.>	<Date>												
		Total production volume											
		Rated peak system capacity											
		Annual investment in DKK											
		Total investment in DKK											

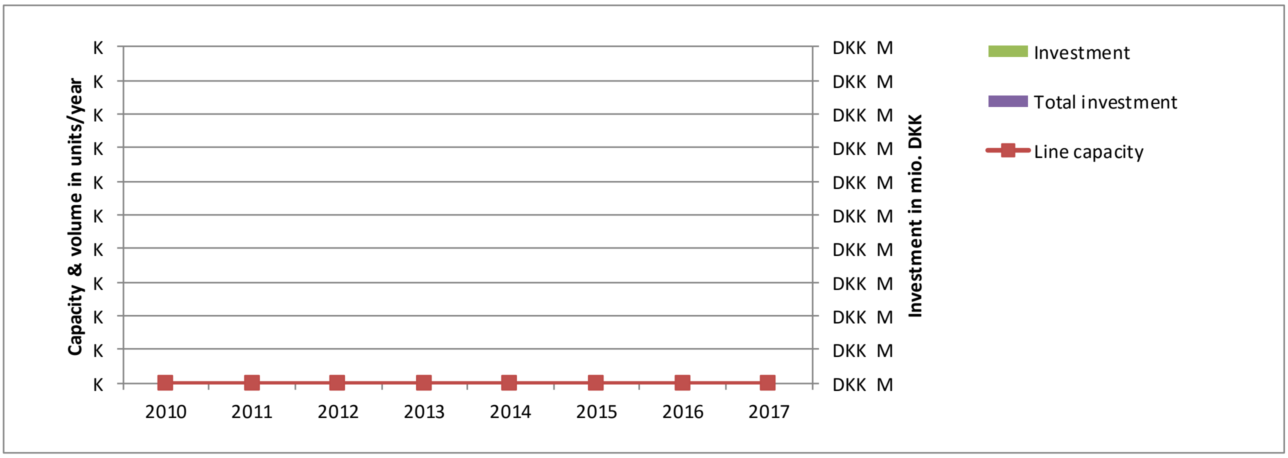


Figure 45 - Template for PCV-4 (Production capacity plan).

Example

An example of the PCV-4 for an assembly line is shown in Figure 46. Due to confidentiality constraints the example shows anonymized data, but it sufficiently resembles the tested production capacity plans to provide an impression of the real world examples.

System info		Output info		Production dates		Past			Present & future				
System name	System ID	Output variant	Intended product	Start	End	2010	2011	2012	2013	2014	2015	2016	2017
Line X	12345	Variant 001	Product family A	05-2007	01-2020	335 K	421 K	657 K	850 K	1.000 K	900 K	700 K	550 K
		Variant 002	Product family A	07-2013					40 K	100 K	200 K	350 K	600 K
Location	Transfer date	Variant 003	Product family B	02-2014						50 K	100 K	250 K	450 K
STX/DK	01-06-2013												
GBJ/DK													
		Total production volume				335 K	421 K	657 K	890 K	1.150 K	1.200 K	1.300 K	1.600 K
		Rated peak system capacity				750 K	750 K	750 K	1.500 K	1.500 K	1.500 K	1.500 K	2.000 K
		Annual investment in DKK				35 M			15 M			6 M	
		Total investment in DKK				35 M	35 M	35 M	50 M	50 M	50 M	56 M	56 M

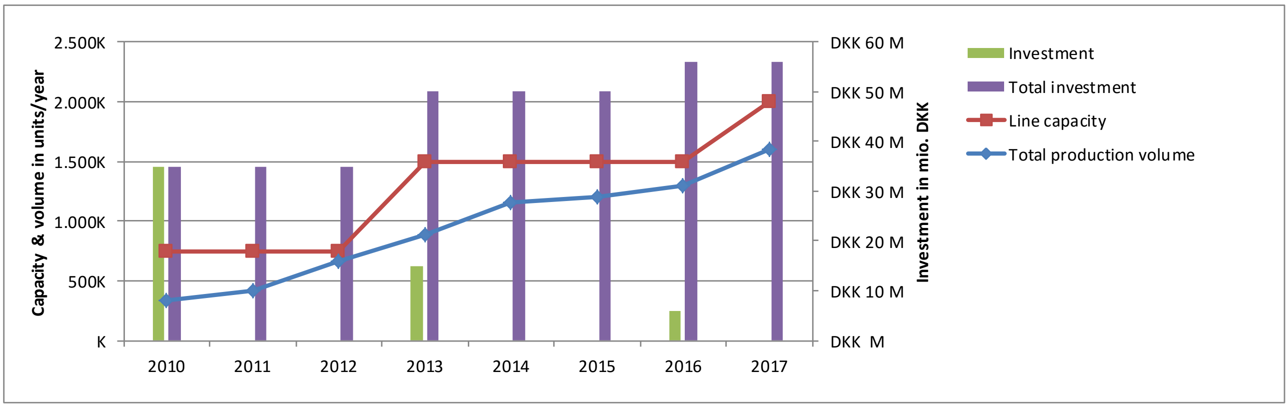


Figure 46 - Example of a production capacity plan for an assembly line.

Remarks on development

The production capacity plan has been developed by modeling of subassemblies for product families. The Plan has been used as a part of a collective overview for a group of production systems (existing, planned and in development) concerned with the production of a family of sub-assemblies for an assortment of circulator pumps. The model kind proved effective not only in describing the capacity plan of individual production systems, but also showed its strength in balancing and distribution of production between the different production systems. This proved especially valuable in the discussion of concerns regarding the capacity requirements of new production systems.

23.5 PCV-5 Change impact roadmap

The PCV-5, Change impact roadmap, frames concerns relating to the necessary changeability or preparedness in the design of the production system, which is the result of changes in technologies or the products to be produced.

What does it do?

The PCV-5 describes the roadmaps of relevant technologies and products and links their design impacts to the roadmap of the production system. The PCV-5 thereby provides stakeholders with an overview of the technologies and products that will or might affect the design of the production. In this way the PCV-5 specifies requirements for the production system's capability to adapt to changes in technology or the product to be produced, and serves to frame a host of concerns regarding the engineering design, investment planning, technology roll-out etc. in relation to the production system.

What is it?

The PCV-5 is a roadmap model that consolidates roadmaps for technology (product technology and production technology), products and the production system(s). Unlike many other roadmaps the PCV-5 is not concerned with typical roadmap information seen in generic roadmaps such as the associated resources, external influences, needed skills or knowhow or deliverables (EIRMA cited in Phaal, Farrukh and Probert, 2001). The focus of the modeling of roadmaps in PCV-5 is on the life-cycle of technologies, products and the production system, and on the existence and timing of the impact on the production system stemming from changes in technology or products to be produced.

The links between the roadmaps show when technology and products to be produced are expected or planned to impact the production system e.g. when new technology is scheduled to be implemented in the production system. The impacts of changes in production technology directly affect the production system, while the impact of changes in product technology are seen indirectly through the changes to the processing required to produce new variants of products. The exact nature and effect of the modeled links are not described in PCV-5 for two reasons:

- Each link may represent a group of design impacts that are in most cases too varied and numerable.
- The information is captured in other model kinds in the architecture description that are capable of describing the changeable aspects of the production system.

What can instead be learned from a link in PCV-5 is that there is an impact that is significant enough to be tracked and communicated to the group of stakeholders. The PCV-5 also links the technologies to their instigating factors i.e. factors instigating the development or use of the technology, to provide a business context for the resulting impact on the production system. This offers a context for the changes to the production system based on the production systems broader role in the company.

Intended usage

The intended usage of PCV-5 includes:

- Provide an overview of the technology that can affect change in the production system
- Provide an overview of the products (operands) that can affect change in the production system
- Provide an overview of when technology or product driven change to the production system might or will occur
- Act as a point of agreement between stakeholders in regards to the long term planning of the production systems evolution and associated investment
- Act as specification of requirements to the changeability and level of preparedness in the design of the production system
- Serve to align the development of technology, products and the production system

It should be noted that the PCV-5 not only serves to provide stakeholders of an overview of the relations between technology, products and the production system. The PCV-5can also serve as a point of agreement between stakeholders in regards to the implementation of technology and products in the production system e.g. what technologies are used in the system and when; what products (based on a certain product technology) should the system be capable of producing; what should be the level of preparedness in the system for technology changes, etc.? Decisions regarding what technologies to use in the system and when this should use should occur are central to a long term design focus for production systems. It is also important in the decisions regarding the roll-out of new technologies and in determining investments in production. This is particularly relevant in the context of ensuring the availability of technologies needed in the production system design.

Introduction to the modeling formalism

The PCV-5, Change impact roadmap, addresses stakeholder concerns regarding the changeability of the production system in relation to changes in the utilized production technology and the products produced. The PCV-5 is focused on aiding stakeholders in the difficult task of determining the necessary levels of preparedness in the design, given the oftentimes non-transparent and uncertain nature of technology development and product development. This means that the PCV-5 is intended to provide an overview of possible influencing changes to the production technology or the products produced throughout the production system life-cycle, and that the PCV-5 should help facilitate discussion, decision making, documentation and communication of the system's capability to adapt to such changes.

Additionally the PCV-5 aids stakeholders in aligning the roadmaps of technology (product and production), products (those of relevance for the production system) and the production system. In this sense the PCV-5 helps stakeholders to discuss the fit between the production system design and the technology and product strategy within the company. This includes:

- Ensuring that production and product technology is available at the appropriate time
- Schedule roll-out of technology to products and production systems
- Schedule start and end of production for products that necessitate change in the system

Technology and product development in most companies is the subject of secrecy and uncertainty. Indeed, observations in the primary case company has demonstrated that information regarding the content, maturity and consequences of technology development and product development is not always available or easily found; sufficiently reliable or informative; or restricted to certain parts or stakeholders of the company. The natural uncertainty and secrecy surrounding new technology development and product development unfortunately means that stakeholders involved in the development and design of production systems are sometimes in the dark in regards to which technologies or products might impact their system. This uncertainty and secrecy manifest in the following ways among other:

- It is not clear what technologies and products will or might impact the production system
- Technology and product responsible stakeholders…
  a) …keep their cards close to the body (do not want to tell too many people what they're working on
  b) …do not want to commit to promises they might have trouble keeping in the future e.g. will the technology work, will it be available at the scheduled time, will it perform as expected?
- Few people might have a full picture of relevant activities such as technology development, product development, ongoing production, financial planning etc.
- Not all information is "committed to paper"
- Many impacts of technology or products can be hard to quantify or qualify, which is why stakeholders might not know if the impact is relevant or if they can commit to saying it will have an impact
- Stakeholders from the different functions of the company have different priorities or desires.
  a) Technology developers might not want to use the technology in a certain production system until the technology is more mature e.g. a system located at a foreign production company.
  b) Internal competition can mean that stakeholder seek to monopolize technology
  c) The uncertainty of new technology can clash with issues of predictability and reliability in the design of production systems.

This makes the task of designing production systems with the appropriate changeability very difficult and means that the architecture of the system does not always sufficiently account for significant changes in the production task or the technological means of production. The PCV-5 is intended to change this by way of providing key stakeholders with a collective means of documenting the relevant planned, expected or potential technology and product changes, and to map the impact on the production system. The PCV-5 addresses the difficulty of this task by doing several things:

- It elicits information from technology development, product development, production system development and production
- It provides a common means of describing the technology development and product development and relating it to the products or production systems of interest.
- It makes the impact of expected or planned technologies and products visible
- It forces stakeholders to commit to the answers they have given to the architects
- It serves as a point of agreement between stakeholders

In general the PCV-5 serves as both an information tool and a means of documenting agreements concerning the implementation of new technology in the company's design of products and production systems. Mixing both product and production into the model is done to facilitate the willingness to use and share information in an environment that is by nature somewhat secretive or at least restrictive about the sharing of information even within the same company.

Focus

The focus in PCV-5 is on determining the existence and timing of impacts to the production system from technology and products. The included technology and products are those that are either objectively determined to affect the production system design or judged by the architecting stakeholder to be of relevance.

Data/information in model

The PCV-5 models the technology roadmap, product roadmap and production system roadmap in a single model, and the model describes links between the roadmaps that represent design impacts.

Unlike many generic roadmaps, the PCV-5 does not focus on resources for projects, deliverables, external influences or specification of necessary skills or knowhow. The focus of PCV-5 is on the identification of technologies, products and their respective impacts on the production system. The exact nature of the impacts are described in other model kinds, that describe differences in processing and system design as a result of the impacting technology or product to be produced. Other roadmaps may exist that support the development and planning of technology development, but these are not included in the production system architecture descriptions. As such the PCV-5 contains the following data and information:

- ID/reference, name and status of technologies or technology development projects
- ID/reference, name and status of products or product development projects
- ID reference, name and status of production systems or production system development projects
- Instigating factors for technology development or implementation
- Presence and timing of design impacts between:
  a) Technology roadmap and Production system roadmap
  b) Technology roadmap and Product roadmap
  c) Product roadmap and Production system roadmap
- ID of stakeholders responsible for the roadmaps and the technologies, products and production systems within them

Type of model

The PCV-5 is a visual model containing, among other things, timeline and tables describing the roadmaps of technology, products and production systems or their related development projects.

Modeling formalism

The PCV-5 consists of three roadmaps (see Figure 47):

- technology roadmap (product and production technology)
- product roadmap (those relevant to the production system)
- production system(s) roadmap

The three roadmaps are linked to show the ultimate impact on the production system and demonstrate which technologies and products will impact the production system, and when this is expected or scheduled to occur. The technology roadmap models both product technologies and production technologies and can reasonably be split into two if so desired: a product technology roadmap and a production technology roadmap. This would be done if product and production technology are referred to entirely separately in the company. In some cases however technology development projects contain developments in both product technology and production technology in which case stakeholders may refer to the technology as one technology, even though it covers both product technology and production technology. In this case the technology or technology project would show an impact on both the products and production system that would need to be tracked, and the technology roadmap would cover both product and production technology.

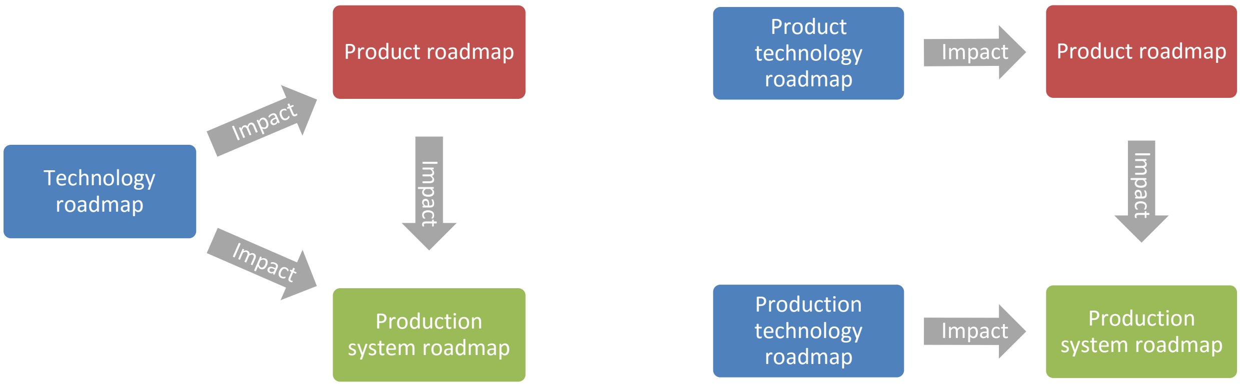


Figure 47 - The principle structure of PCV-5 showing a combined technology roadmap (Left) or separated technology roadmaps (Right). Impacts are mapped between roadmaps.

Technology roadmap

The technology roadmap models the product or production technologies of relevance for the production system or products to be produced. This includes technology that is either implemented directly in the production system or technology that gives rise to new products to be produced by the production system. The technologies may be represented by projects associated with their development.

The technology roadmap consists of two elements: 1) a table of technologies or technology projects, and 2) a timeline diagram for the technology's life-cycle or project timeline (see example in Figure 48). The technology table lists the technologies, along with their status/maturity and references to the motivating factors precipitating the development or deployment of the technology. The following information is included in the table:

- **Number:** A reference number for the technology specific to the roadmap, shown as a colored circle with a number in it. These are used as references for the impacting technology in the other roadmaps of PCV-5 Colors can be chosen to signify the different organizational units responsible for the technology.
- **ID:** ID or other reference for the technology or project
- **Name:** Name of the technology or project

- **Resp.:** ID or name of responsible stakeholder. Multiple stakeholders may be added if different responsibilities need to be mapped.
- **Motivators:** Reference to the motivators for the technology e.g. text or representative symbols
- **Status:** Reference to the status/maturity of the technology e.g. text or symbols. (In the tested cases an abstract representation of the maturity or development models used in the company were employed for easy visual recognition)

Please note that the syntax for all of the elements is company specific, and that a reference to the contact responsible for gathering the roadmap data is also included.

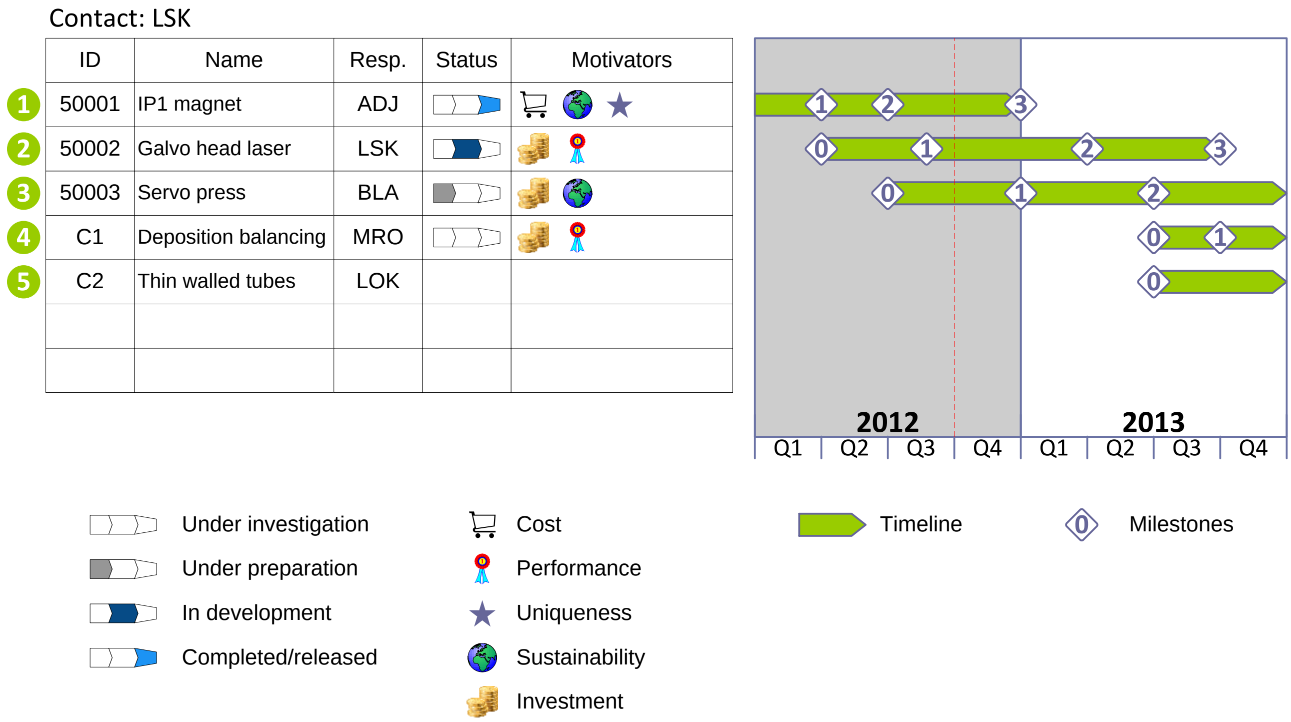


Figure 48 – Example of the technology roadmap in PCV-5. The roadmap may be sub-divided by different organizational units and will utilize Technology motivators, Technology status indicators and Development process models specific to the company.

The technology motivators shown for each technology represent a business justification for developing or using the technology, and the technology status indicates the maturity of the technology or the current status. Experience from the case company has shown that different organizational units may employ different models or ratings for defining the maturity of technologies. For the sake of communication between stakeholders it is advised to use a common model as reference for the maturity of technologies if possible for example an adaptation of the Technology Readiness Levels defined by NASA (NASA, 2013). It could also be as simple as the status shown in the example in Figure 48.

The timeline diagram shows the timeline of the technology project timeline or the technology life-cycle. Indicators for milestones in projects or transitions in the life-cycle stage of the technology are included to provide an overview of past and future transitions in the status of the technology or project.

Product roadmap

The product roadmap models the planned or expected output operands (products) of the production system, either individual variants, groups of variants or archetypes of variants (as defined in PVV-1). The operands may be represented by the projects associated with their development.

The roadmap consists of two elements: 1) a table of operands or their associated development projects, and 2) a timeline diagram for the operands' life-cycle or project timeline (se Figure 49). The technology table lists the operands, along with references to the technology or technology projects that are planned or expected to have an impact on the operand design. The following information is included in the table:

- **Number:** A reference number for the operand specific to the roadmap, shown as a colored circle with a number in it. These are used as references for the impacting operand in the production system roadmap. Colors can be chosen freely to signify different groupings or associations of operands.
- **ID:** ID or other reference for the operand or project
- **Name:** Name of the operand or project
- **Site:** Reference to the company site where the operand is developed or managed
- **Resp.:** ID or name of responsible stakeholder. Multiple stakeholders may be added if different responsibilities need to be mapped.
- **Technology relation:** The technology numbers are used to reference the technologies or technology projects that are expected or planned to impact the operand.

The timeline diagram shows the timeline of the technology project timeline or the technology life-cycle. Indicators for milestones in projects or transitions in the life-cycle stage of the technology are included to provide an overview of past and future transitions in the status of the technology or project. Comparing related timelines of the product roadmap and the technology roadmap thus allows for alignment of the

Contact: KWK

	ID	Name	Resp.	Status	Technology relation
1	10001	Product A - Small	KWK		1
2	10002	Product A - Medium	KWK		1
3	10003	Product A - Large	-		1 5
4	20001	Product B	OKJ		1
5	20002	Product B – OEM	HEK		5
6	30001	Product C	TTJ		5

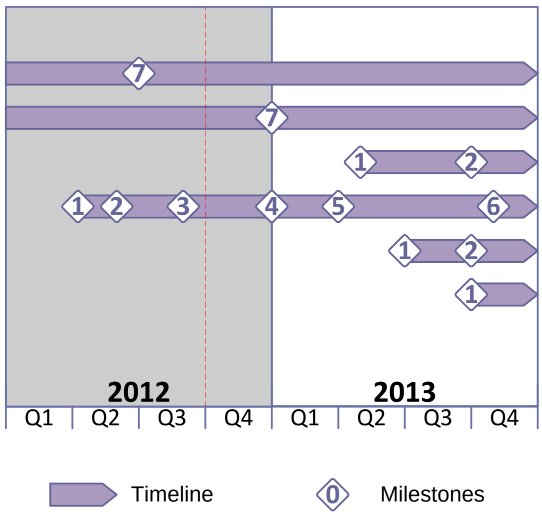


Figure 49 - Example of the product roadmap in PCV-5. The roadmap may be sub-divided by different organizational units and will display the relations to the technology roadmap. The development process model is specific to the company.

Production system roadmap

The production system roadmap is used for two purposes: 1) documenting the planned or expected life-cycle of the production, and 2) documenting the impacts of technology or products in relation to the production system life-cycle. The production system roadmap models the production system(s) covered by the architecture description. If the architecture description covers a group or family of production systems, then multiple systems are included in the roadmap.

The roadmap consists of two elements: 1) a table of production systems, and 2) a timeline diagram for the production systems' life-cycle (see Figure 50). The table lists the production systems, along with references to the technologies and products (operands) that are planned or expected to have an impact on the production system. The following information is included in the table:

- **Number:** A reference number for the production system specific to the roadmap. Future production systems that are only anticipated but not confirmed have an A (for Anticipated) preceding their reference number. They are given a new number when finally confirmed.
- **ID:** ID or other reference for production system
- **Name:** Name of the production system
- **Resp.:** ID or name of responsible stakeholder. Multiple stakeholders may be added if different responsibilities need to be mapped.
- **Technology relation:** The technology numbers are used to reference the technologies or technology projects that are expected or planned to impact the production system.
- **Product relation:** The product numbers are used to reference the operands that are expected or planned to impact the production system

Contact: KWK

ID	Line name	Resp.	Status	Product relation	Technology relation
DK009	DK - UP 1 line	SSL	Retired	1	
DK010	DK - Flexline	SSL	Operation	1 2 3	1 2
HU003	HU - M/L line	PBJ	Run-in	2 3	1 3
US003	US line 3	TSO	Concept	4	1 2 4
US004	US line 4	LHA	Research	5 6	4

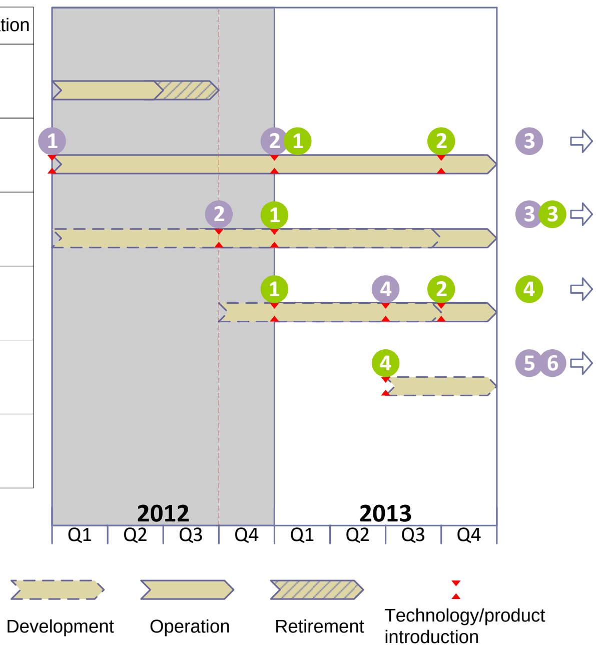


Figure 50 - Example of the production system roadmap in PCV-5

The timeline diagram shows the timeline of the production system life-cycle. The impacting technologies and operands are mapped along the timeline to show when the impact occurs. The life-cycle stages of the production system are indicated by changes in the color or outline of the timeline. Comparing related timelines of the production roadmap, product roadmap and the technology roadmap thus provides an overview of the impacts on the production system, and allows for alignment of the three roadmaps.

If a finer resolution of the impact on the production system is needed, it is possible to model the sub-elements of the production system (provided that these are known). The PCV-5 will then link technologies and products to the sub-elements of the production system and show which elements of the production system will be impacted by the technology or products.

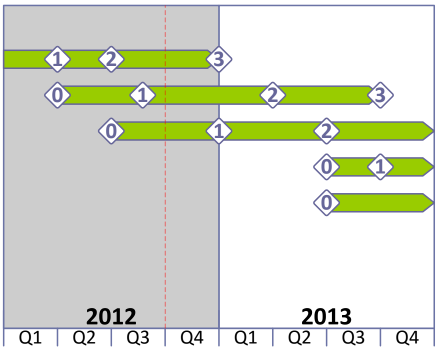
Example

An example of the PCV-5 for a family of assembly lines is shown in Figure 50. Due to confidentiality constraints the example does not represent a real life example of the PCV-5, but it sufficiently resembles the tested models to provide an impression of the real world examples.

Technology roadmap

Contact: LSK

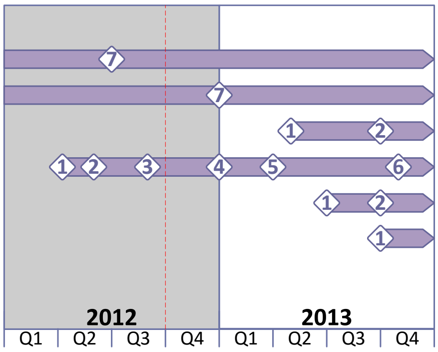
	ID	Name	Resp.	Status	Motivators
1	50001	IP1 magnet	ADJ		
2	50002	Galvo head laser	LSK		
3	50003	Servo press	BLA		
4	C1	Deposition balancing	MRO		
5	C2	Thin walled tubes	LOK		



Product roadmap

Contact: KWK

	ID	Name	Resp.	Status	Technology relation
1	10001	Product A - Small	KWK		1
2	10002	Product A - Medium	KWK		1
3	10003	Product A - Large	-		1 5
4	20001	Product B	OKJ		1
5	20002	Product B – OEM	HEK		5
6	30001	Product C	TTJ		5



Production system roadmap

Contact: KWK

ID	Line name	Resp.	Status	Product relation	Technology relation
DK009	DK - UP 1 line	SSL	Retired	1	
DK010	DK - Flexline	SSL	Operation	1 2 3	1 2
HU003	HU - M/L line	PBJ	Run-in	2 3	1 3
US003	US line 3	TSO	Concept	4	1 2 4
US004	US line 4	LHA	Research	5 6	4

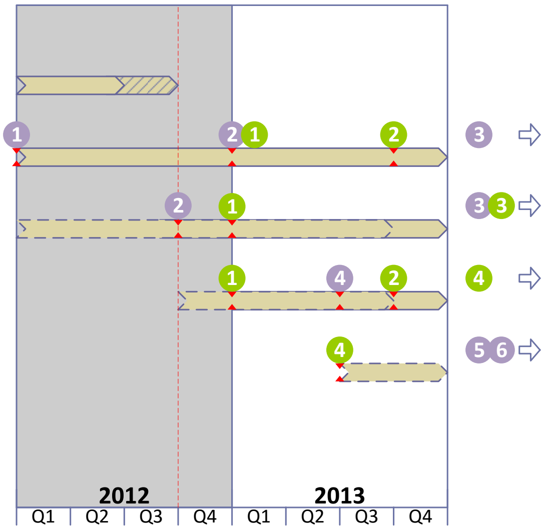


Figure 51 - Example of PCV-5 including fictitious data but inspired by the tested models.

Remarks on development

The PCV-5 as it is presented in its final form represents the condensed modeling formalism tested throughout both platform development projects for production systems and technology management projects in the company. The roadmap was found to provide stakeholders with a more focused overview of relevant technology developments compared to the usual roadmaps that cover an entire R&D organization. Having a focused overview enable the stakeholders from technology development and production system development enter into a discussion of the possible impacts between their domains and the production systems that are in the development pipeline. The PCV-5 has not been tested for single production systems, and as such it cannot be said if it will be equally useful, when the architecture description covers only a single production system.

24 Technical System Viewpoint (TSV)

The Technical System Viewpoint (TSV) addresses the design related concerns of system architects, engineers, and other stakeholders involved in the design of the technical system of the production system. The viewpoint supports the design activities involved in designing a technical system in compliance with the capability requirements specified in the Production Capability Viewpoint. On a long term basis the viewpoint also supports documentation of the system evolution as the system naturally changes throughout the life-cycle.

As has been said the production system is defined as a transformation system consisting of five operators (TS, HuS, InfS, MgtS, AEnv). PSAF at the present time does not cover all the operators, though it should be expanded to do so. As such the PSAF could in future have a HSV (Human System Viewpoint), ISV (Information System Viewpoint, MSV (Management System Viewpoint), ESV (Environmental System Viewpoint), or viewpoints describing more than one system or their relations.

The TSV is intended to provide a description of the constituent design of the technical including how it relates to the required capability specified in the PCV. The model kinds that make up the viewpoint offer a description of the system design that includes many of the design aspects that are traditionally associated with the architecture of technical systems (c.f. section 9 and 10) i.e. it describes the system elements, structures, allocation relations and changeability. The description of the Technical System offered by the TSV not only addresses the internal constituent design of the system, it also describes the relations to the other operators of the production system, the general environment of the production system and the technical process taking place in the production system. The PCV thus allows for describing the system design and enable communication of the design between stakeholders. Use of the TSV allows these stakeholders to answer questions such as:

- What are the constituent elements of the technical system and their characteristics?
- What is the variety of the system elements?
- What technical solutions implement functionality (production processes)?
- How is the system structured?
- How is the changeability of the system realized in the design of the system?
- What configuration options are possible in the system design?
- What are the interfaces of the system both internally and to the external surroundings?

In general the TSV frames concerns regarding the engineering design of the technical system in the production system, the design of the other operators (Human System, Information System and Management system, Active & Reactive Environment) is not included, but part of the relations to these systems are included in the description of the technical system. The TSV consists of the following model kinds that will be detailed further in separate sections:

Table 4 – List of model kinds in the Technical System Viewpoint

Model	Name	Description
TSV-1	Concept diagram	The concept diagram describes the basic concept behind the design of the production system
TSV-2	Interface diagram	The interface diagram describes the interfaces between elements of the system (organs and parts).
TSV-3	Configuration diagram	The configuration diagram describes the different configuration options possible within the architecture.
TSV-4	Technical System Master Plan (TSMP)	The TSMP provides an overview of the variety of a technical system or family of technical systems as perceived by users and designers of the system.

Each of the model kinds included in the viewpoint are described in the following sections.

24.1 TSV-1 Concept diagram

The TSV, Concept diagram, provides stakeholders with an understanding of the basic design concept of the production system, and helps address stakeholder concerns regarding the engineering design of the production system's technical system.

What does it do?

The TSV-1 describes the concept behind the system's design. The concept description may cover a mix of design elements including but not limited to:

- Process design
- Functionality
- Technologies or technical designs
- Qualitative/functional structure for elements
- Quantitative/physical structure for elements
- Organ/sub-system design
- Part design

The content of the TSV-1 is decided by the architecting stakeholders, and should provide a high level description judged suitable enough to convey the main elements of the system design.

What is it?

The TSV-1 is a diagram of unspecified format with a high level of abstraction. The TSV-1 does not specify the concept diagram to follow any specific format, but does specify that there should be a visual component to the chosen modeling format, since this is judged appropriate to convey abstract ideas to stakeholders with a diverse technical background.

Intended usage

The intended usage of TSV-1 includes:

- Documentation of the design concept for the technical system
- Design support in the design process
- Communication with both technical and non-technical stakeholders.
- Sub-dividing the design task

Introduction to the modeling formalism

The TSV-1, Concept diagram, provides a description of the concept underlying the system design. The TSV-1 can be seen to provide a high level description of the system-design to be realized in the engineering design process. The concept is modeled using appropriate means, and may or may not follow a modeling formalism. The model can be used both as a design tool for communication with stakeholders and as a documentation of the evolution of the system throughout the system life-cycle.

Data/information in model

The type and amount of data in the TSV-1 depends on the necessary information that needs to be conveyed in order to adequately describe the system concept, and it depends on the means of modeling chosen by the architecting stakeholder. Among the commonly included information is the following:

- System functionality
- Technology selection
- Technical design solution
- System structure
- Element design characteristics (dimensions, weight, cost, source, etc.)

Type of model

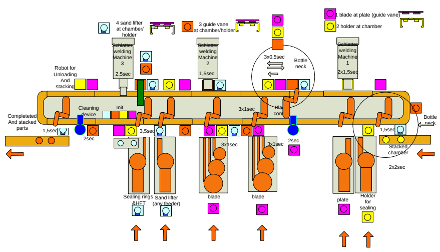
The TSV-1 is primarily a visual model, but PSAF does not specify a specific means of modeling the concept. Stakeholders are free to choose an appropriate means of modeling the concept, provided there is a visual aspect to the model. The visual aspect is chosen because, as previously mentioned, it is beneficial in facilitating communication between stakeholders, as shown by for example (Alabastro et al., 1995).

Modeling formalism

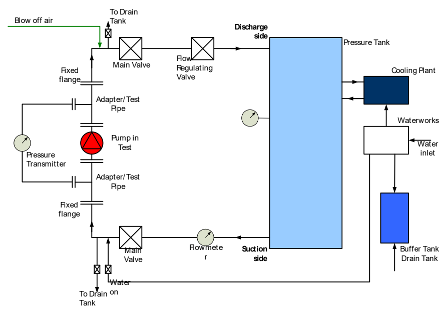
Although the TSV-1 does not mandate the use of a specific means of modeling, the following modeling means are suggested to be used in TSV-1 (see Figure 52). All of the suggested modeling means were encountered in the case work, and all except for 3D models have been tested as a means of modeling the concept of production systems as part of the architecture description:

- **Freestyle drawings:** Freestyle drawings and supporting text. These may employ a company specific set of symbols, shapes or terminology for design concepts often used in the company. Typically architecting stakeholders use a set of shapes or symbols that are commonly understood by other stakeholders within a company.
- **Schematics:** Different schematics may be used that follow standardized notation formats e.g. Piping & Instrumentation diagrams or electrical diagrams. Standardized notation formats may be modified to include company specific symbols.
- **2D layout diagrams:** High level layout diagrams with approximate proportionality of dimensions and/or layout.
- **3D models:** 3D models with approximate proportionality of dimensions and layout.
- **Generic Production Flows:** A modeling notation that visualizes the processing flows for variants and links the processing steps to equipment (Mortensen, Hansen, Hvam and Andreasen, 2011; Ravn, Guðlaugsson and Mortensen, 2014).

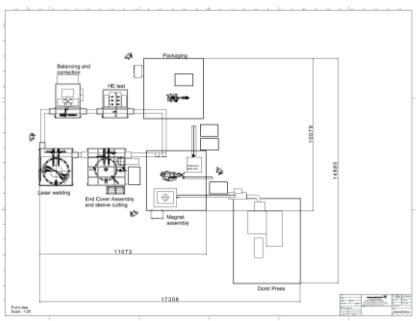
Drawings



Schematics



2D layout diagrams



3D models

Generic Production Flows

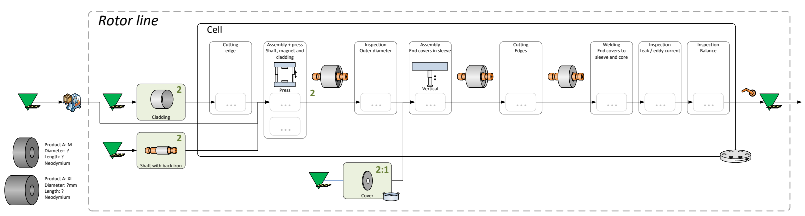


Figure 52 - Suggested means of modeling in TSV-1

Example

An example of the PCV-5 for a family of assembly lines is shown in Figure 53. The example uses sparse text and symbolism because the system design is familiar to most relevant stakeholders, and similar systems are already in operation within the company.

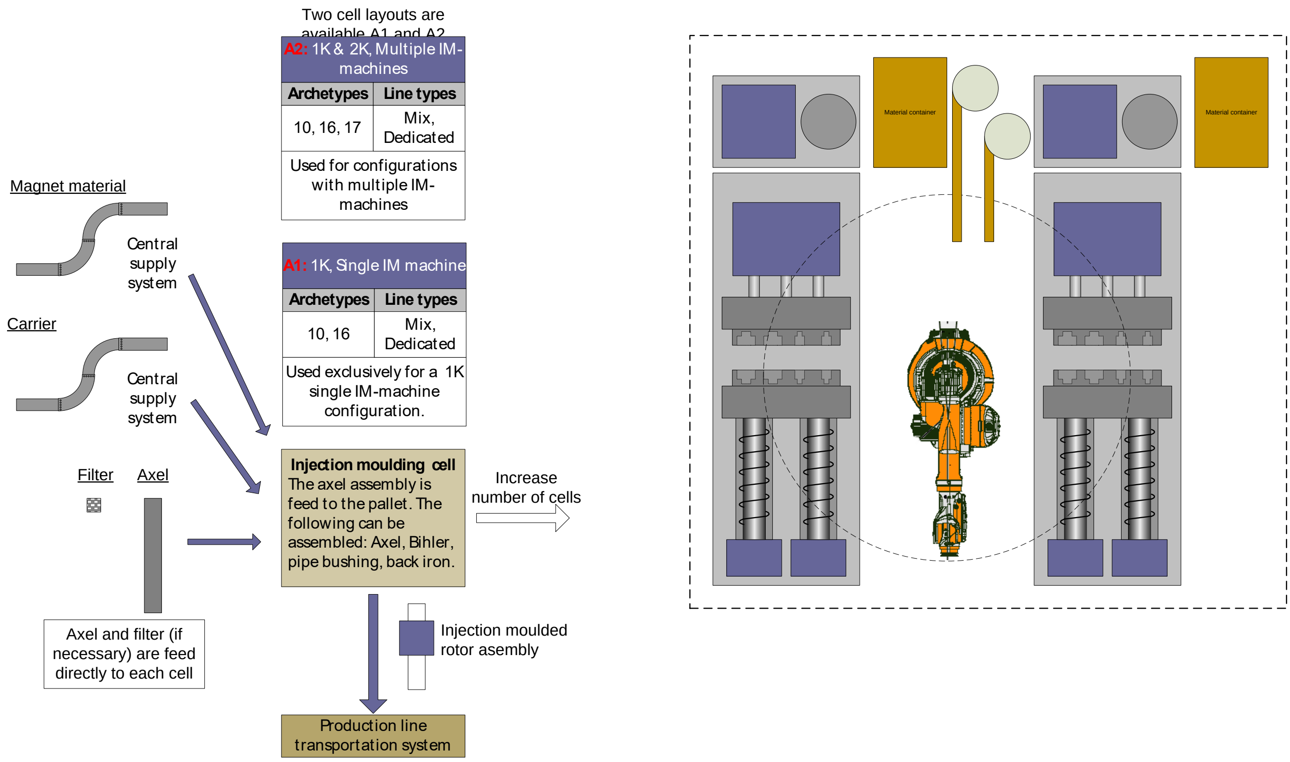


Figure 53 - Example of a simple concept drawing for an injection molding cell with two different setup types

Remarks on development

The case work at the case companies has included testing of Drawings with text, 2D layout diagrams and Generic Production Flows. It has not been possible to draw any conclusions on which modeling means to prioritize. All of the means have their own advantages and disadvantages that are highly dependent on the knowledge of the stakeholders, the system to be designed, the design task and the amount of information to be conveyed. The PCV-1 therefore leaves it up to the architecting stakeholders to choose a means of modeling the system concept that is appropriate to the task. The Generic Production Flows are currently used in Grundfos within product development, and have been shown to provide a good description of production concepts. It is recommended that this modeling formalism be prioritized in the selection of a means of describing the concept in TSV-1.

24.2 TSV-2 Interface diagram

The TSV-2, Interface diagram (IFD), addresses concerns regarding communication and collaboration between stakeholders in a multidisciplinary development process; the successful integration and interaction of system element; the connection between functional system design and physical part design; and the control of design ownership and responsibility for functional sub-systems, parts and interfaces in the Technical system. The TSV-2 is intended to bridge the gap between system oriented engineering and part oriented engineering by explicitly describing the design of the Technical system from a system and part perspective.

What does it do?

The TSV-2 provides a structural description of the Technical system that allows for mapping between the physical part design and the functional system design of the Technical system, and allows for tracking and managing of systems, parts, interactions or interfaces in the Technical system and to the system's surroundings.

The TSV-2 models a structure consisting of functional systems and/or parts, where the structure is defined by the interactions or interfaces between the elements. Both systems and parts may be subdivided to a degree sufficient enough for assignment of responsibility and for tracking and management of the necessary systems, parts, interactions or interfaces. The structure described in the TSV-2 can act as a reference for other viewpoints on the system, where, by applying filters to the structure, the stakeholders can investigate alternative structures and gain an understanding of the different structuring requirements of different technical domains, life-cycle stages, etc.

What is it?

The TSV-2 is an object oriented modeling formalism that models the Technical system's functional systems and parts, as well as interactions and interfaces between them in the form of a block diagram. The diagram describes a structure of object classes representing either the technical system's generic organ structure (functional system structure), generic part structure, or any combined structure thereof. The elements of the structure are either functional systems (also known as organs or functional carriers) or parts, and the relations between elements that define the structure are either the elements functional interactions or the interfaces through which the interactions occur.

The TSV-2 represents an adaptation of the Interface diagram modeling formalism presented by (Bruun, Mortensen and Harlou, 2014) and its inspiration the Generic organ diagram (Harlou, 2006) with some modifications and extensions to the syntax and semantics .

Intended usage

The TSV-2 is intended to be used throughout the life-cycle of the production system both as active design and documentation support, and as a means of keeping track of design responsibilities and dependencies. The intended usage of TSV-2 includes:

- Interdisciplinary communication and collaboration
- Designing the system from different viewpoints e.g. the functional (system) and physical
- Mapping between technical domains
- Managing interfaces
- Assigning and keeping track of responsibilities

Typically the TSV-2 will be printed as a large poster around which stakeholders can gather to discuss the system structure, but the model also works in a digital context.

Introduction to the modeling formalism

Developing production systems, and by extension the technical system, is a process associated with many challenges that are dependent on the stakeholders' ability to view the system's structural design from many viewpoints and exercise careful control of the system interfaces. Some of the challenges encountered include:

- Communication and collaboration between stakeholders from different domains
- Integration of system elements designed and built by different organizational entities or external suppliers
- Ensuring performance in the different life-cycle stages
- Assignment and control of ownership and responsibility for elements of the design

The TSV-2 seeks to address these challenges by allowing stakeholders a multi-structural view of the system and its interfaces in relation to a basic structure that describes the connection between the functional sub-systems and parts of the technical system.

The structure of the TSV-2 allows for the explicit modeling of interactions or interfaces between functional elements or physical elements of the Technical system, thus providing a better basis for the effective management of interfaces in the technical system. This is important both in the initial design of a new system and in the continued management of the Production system. The effective management of interfaces and system interactions is crucial in the design of production systems whose design is prepared for the introduction of new technology, equipment or products to be produced as specified in the production system's capability.

The TSV-2, Interface diagram, is an object oriented modeling formalism consisting of a block diagram that models the Organ structure and Part structure of the Technical system. The TSV-2 is based on the modeling formalism of the Interface diagram (IFD) (Bruun and Mortensen, 2012; Bruun, Mortensen and Harlou, 2014) and its inspiration the Generic organ diagram (Harlou, 2006). The TSV-2 version of the Interface diagram makes some alterations to the syntax and semantics presented by (Bruun, Mortensen and Harlou, 2014) but maintains the key elements. In the following descriptions of the modelling formalism the abbreviation IFD will be used to refer to the modeling formalism described by (Bruun, Mortensen and Harlou, 2014), and TSV-2 will be used to refer to the modified modeling formalism used in PSAF.

Modeling formalism

The TSV-2 allows the architecting stakeholders a multi-structural perspective on the Technical system that enables them to successively decompose the system into its constituent elements and map between the functional elements of the system and the physical elements that realize the functional elements. The reason the architecting stakeholders must be able to work from a multi-structural perspective, lies in the nature of designing a Technical system. Within the theory of domains, which forms the basis of the research, the design of a technical system (product) involves the design of three systems that represent three views on the system: the Activity system, Organ system and Part system. The activity system represents the transformation process that the Technical system conducts in conjunction with the other operators of the transformation system. The Organ system describes the product as a system of organs, where the organs are the elements that realize the functionality of the technical system through their effects on the operands. (Hubka and Eder, 1988, p.77) defines an organ as *"a system that realizes a given internal function of a technical system"*, and they are also sometimes referred to as function carriers, functional units, functional elements or functional systems. The part system describes the product as a system of parts, where parts are the elements that realize the organs' mode of action by their physical state and interaction. Parts as defined in the theory domains are either material elements or material surfaces, and they are classified by four characteristics: Form, Material, Surface quality and Dimension (Andreasen, Howard and Bruun, 2014, p.9). The design of the Technical system can be seen as a synthesis of these three systems (see Figure 54), and PSAF must among other enable the description of the resulting system structures. The description of the Activity system is done in the PCV-2, Process flow diagram, which describes the process structure of the technical system, while the organ system and Part system is what is described by the TSV-2 (see Figure 55). The TSV-2 is intended to not only describe the Organ and Part systems, but to enable the architecting stakeholders to map between the two to maintain a multi-structural perspective on the system design i.e. a functional system perspective and a physical part perspective.

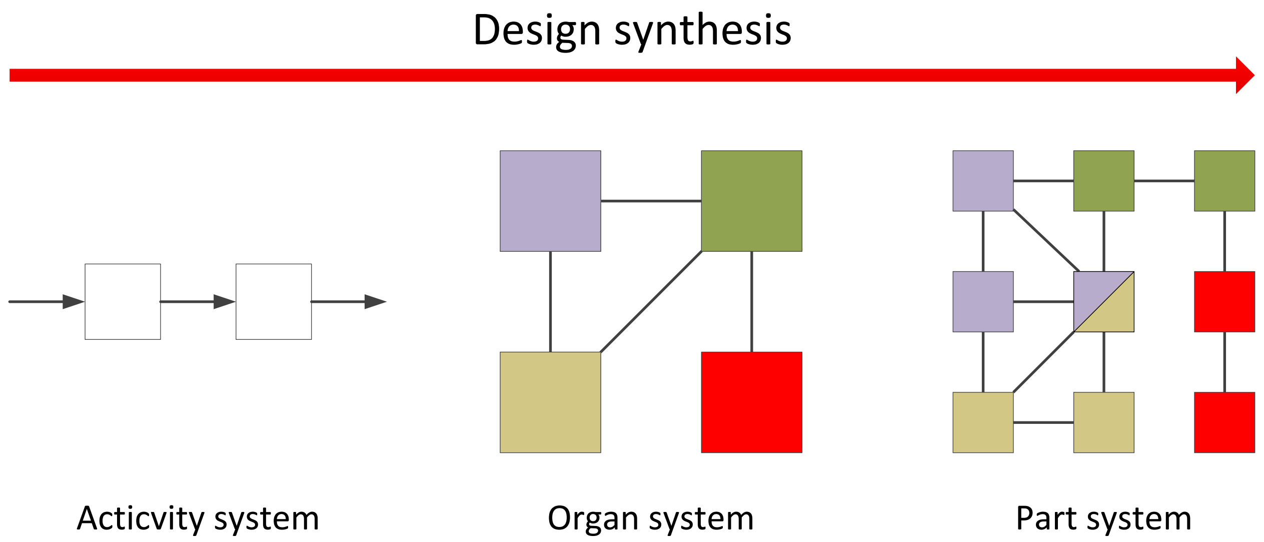


Figure 54 - The basic model of design synthesis for technical systems. The process concerns the design of a process system, functional system and part system described by the three structures of the figure.

The description of the three structures of the systems could reasonable be done in three separate models, and indeed the PCV-2 and the Generic organ diagram presented by (Harlou, 2006) already represent two such models for describing the Activity structure and Organ structure respectively. The TSV-2 however models both the Organ structure and Part structure in a single diagram, in order to bridge the gap between system focused engineering and part focused engineering. The design of the organ system and part system consists of a gradual detailing of the parts that implement the organs, which means that there are times where the organ and part structure are not fully defined. The TSV-2 bridges this gap, by modeling the Organ and part structure in the same diagram. The TV-2 allows for modeling of any combination of the two structures, where on one end of the spectrum the modeled structure is a pure Organ structure, and on the other end of the spectrum it is a pure part structure (see Figure 55). Whether or not the structure falls towards one end of the spectrum or the other can depend on both the maturity of the design and the focus of the architecting stakeholders in the generation of the architecture description.

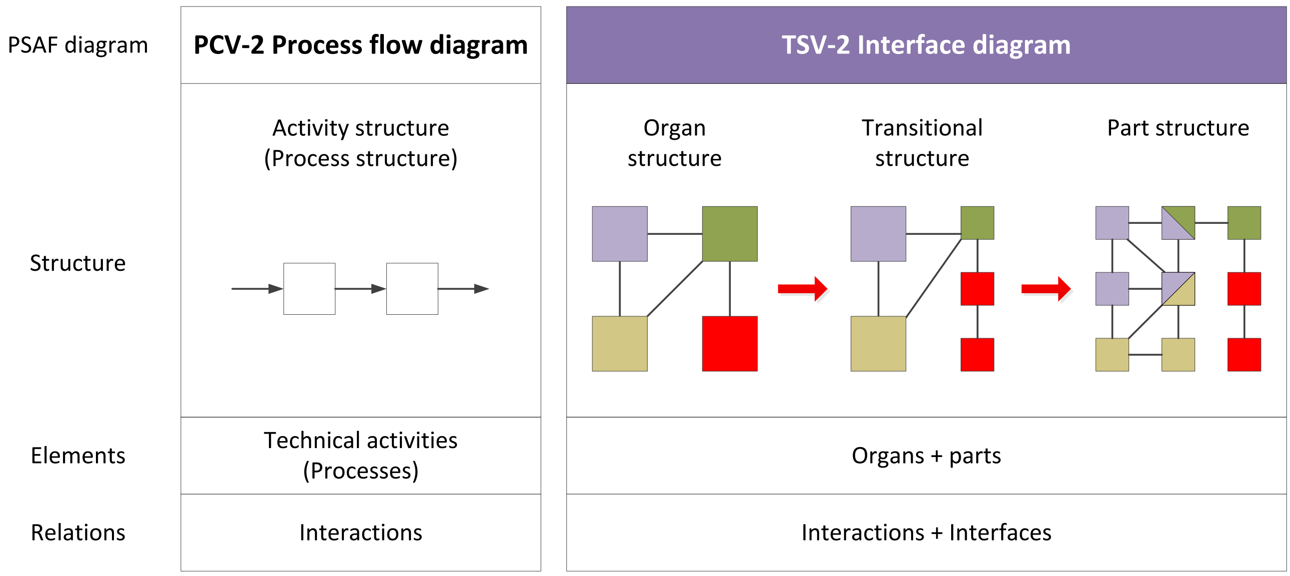


Figure 55 - Description of the three systems of the Technical system within PSAF.

The elements of TSV-2

The main elements of the TSV-2 are referred to in the IFD as Key components (IF components in earlier publications) and they represent the classes of organs and parts in the system. It should be noted that TSV-2 in its current development does not cover non-physical parts, but it is believed that the TSV-2 can be expanded to include non-physical parts of the technical system in the future, although this has not been tested in this research project. Non-physical parts could for example be software and firmware existing as a software package loaded onto the equipment e.g. a disk image copied to the equipment. The key components have different states and characteristics in relation to the system design and life-cycle. The IFD and its inspiration the Generic organ diagram describe Key components as being characterized by four aspects, but neither of the two modeling formalisms include all four:

Design sharing: In architectures covering families or groups of production systems the Key components may be re-used/shared between systems. Key components that are shared/reused between system designs are called Standard designs and they are a class of design units.

Design status: The key components may either be existing or future designs that will be designed in the future and need to be considered in the initial design of the complete system

Variety: More than one variant of a key component may exist.

Configuration: Key components may be present or not present in the system i.e. they can be included or omitted from the system to acquire the desired functionality. The Interface diagram formalism describes such Key components as “optional”.

It should be noted that the IFD does claim to model design sharing in the system by way of modeling modules, which are defined as encapsulations of Key components, and are equated with standard designs (Bruun, Mortensen and Harlou, 2014, p.63). However this is not correct as the theoretical foundation for the IFD and its predecessor the Generic organ diagram allows for the definition of modules that are non-standard. The IFD is based on the Generic organ diagram (Harlou, 2006), and must be expected to follow the same definitions of modules and standard designs therein. Within the Generic organ diagram design units are defined as *“a function, organ, part or an encapsulation of a group of these”* (Mortensen and Andreasen, 1996) cited in (Harlou, 2006, p.79). Standard designs are defined as *“a design unit that complies with one or more product families that will be developed over time. Furthermore, the standard design has to comply with the rules: Decision of re-use, documentation and responsibility”* (Harlou, 2006, p.80), and modules are defined as “one or more design units that are encapsulated into a module and that comply with the module drivers” (Harlou, 2006, p.81). The module drivers are defined by (Ericsson and Erixon, 1999) and summed up by (Harlou, 2006) as:

> *“Carry over – They are reasons that a technical solution should be a separate module since the solution can be carried over to future generations of the product.*
>
> *Technology evolution – Technical solutions that go through a technology evolution during the product’s life cycle should be separated into a module. This might enable update of the module without updating the entire product.*
>
> *Planned product changes – There are reasons that a technical solution should be a separate module because it is the carrier of properties that will be changed according to a decided development plan. These changes are developed in-house or by sub suppliers.*
>
> *Technical specification – Technical solutions that are often influenced by variations in technical specification (different: function, size, torque, etc.) can with advantage be separated into a module.*
>
> *Style – Some parts of the product might be strongly influenced by fashion or trends. It can be beneficial to isolate these parts into a module in order to differentiate the appearance of the products.*
>
> *Common unit – Parts that are identical in all products are candidates for “common unit” modules. A common unit is used across several products.*
>
> *Process and/or organization re-use – Parts of a product that require the same production processes can be clustered into a module. Such clustering might improve the efficiency of the production.*
>
> *Separate testing – The possibility of separate testing of each module before assembly might improve quality. This is mainly due to the reduction of feedback times.*

Supplier offers black box – Sub suppliers might be suitable for development and manufacturing of modules. This implies that the vendor takes the manufacturing, development and quality responsibility.

Service and maintenance – Parts exposed to service and maintenance may be clustered together to form service modules.

Upgrading – Designing modules that allow upgrading of the product, offer customers the possibility of changing the product in the future.

Recycle – There are reasons that this technical solution should be a separate module because of recycle issues, e.g. to isolate highly polluting material." (Harlou, 2006, pp.64–65)

From these definitions it follows that both modules and standard designs are a sub-class of design units (see Figure 56) and that standard designs are modules that comply with the three rules of standard designs (Decision of re-use, Documentation and Responsibility). This means that all standard designs are a form of modules, but not all modules are a form of standard designs, and you could reasonably refer to a standard design as a standard module. The Interface diagram (Bruun, Mortensen and Harlou, 2014) therefore does not model the design sharing of organs or parts across variants of technical system simply by modeling modules.

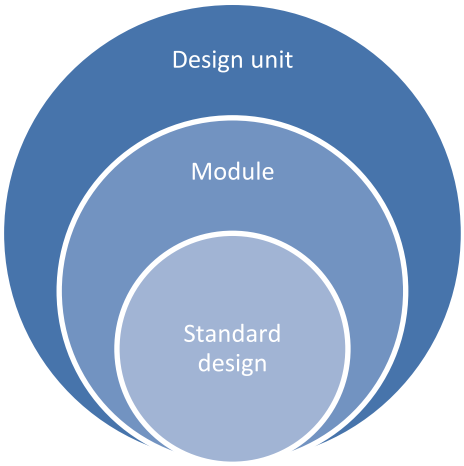


Figure 56 - The class relationship for design units (functions, organs, parts), modules and standard designs.

The TSV-2 therefore merges the characterizing aspects of the IFD and Generic organ diagram to use all four descriptive factors as seen in Table 5:

Table 5 - Descriptive factors in the Generic organ diagram, Interface diagram and TSV-2

	Generic organ diagram	Interface diagram	TSV-2
Design sharing: Shared or unique	X		X
Design status: Existing or future	X	X	X
Design variety: One or more variants	X	X	X
Configuration: Always present or optional		X	X

It should also be noted that the TSV2 suggest using the configuration aspect of Key components to describe life-cycle differences in the system configuration. A Key component that is indicated as optional can be one that is scheduled or expected to be added to the system at a later point in the life-cycle and as such the configuration factor is especially useful in describing not just the configuration differences between variants of Technical systems but also the changeability of a given system.

Key components are modeled in TSV-2 as blocks in the diagrams. The blocks represent the classes of the key components in a fashion similar to the object modeling in PCV-1, where nodes represented classes of operands. The four characteristics of the Key components are symbolized by differences in the color and outline of their representative blocks (see Figure 57). The characteristics are represented as follows:

- *Design sharing:* Key components with a re-usable/shared design are represented by a white block, whereas a Key components with a unique design i.e. not shared/reusable by multiple technical systems is represented by a grey block.
- *Design status:* The design status of the Key component is represented by the line surrounding the blocks. An existing design for a Key component is represented by a full line. Key components that are to be developed in the future are represented by a dashed line.
- *Design variety:* The existence of more than one variant of a Key component is represented by placing a block offset behind the block representing the object class.
- *Configuration:* Key components that are configurable (present or not present) are given a diagonal line through the block.

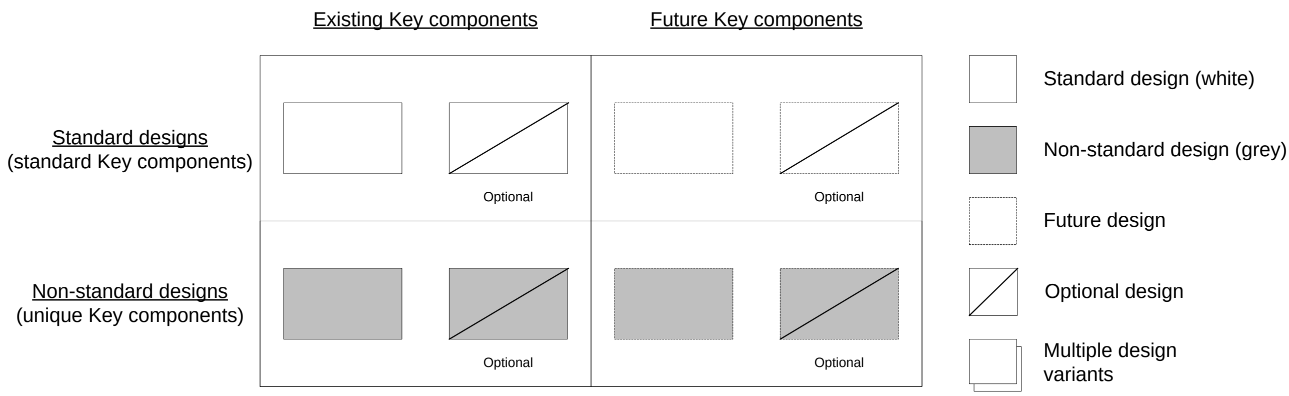


Figure 57 - Representations of Design units

Each block for a Key component can be given the following information:

- Mandatory information
    - Name of Key component
    - Reference Designation (see Part 5 of dissertation)
    - Cardinality of the class (how many Key component in the structure)
- Optional information
    - Responsible stakeholder
    - Primary system affiliation (the functional sub-system that holds the design ownership of the Key component)

- Secondary system affiliations (the functional sub-systems that hold requirements to the Key component but not ownership of the design)

All Key components representing parts in the diagram are a constituent of at least one functional sub-system of the Technical system, or constitute a functional sub-system themselves. The Primary system is the sub-system that determines the design of the part, whereas secondary systems are those sub-systems that only pose requirements to the part.

Interactions & interfaces

The structure of the diagram appears as interactions and/or interfaces between Key components are added. Interfaces and interactions between Key components are represented by lines connecting the blocks. Interfaces and interactions between Key component can be defined by many different means such as the technology used or the nature of the interaction (e.g. transfer of material, energy or information). The TSV-2 models the interfaces seen in Figure 58, but allows for stakeholders to classify interfaces by alternative schemes. Note that TSV-2 models the interfaces for operand exchange as a separate interface or interaction to distinguish it from the exchange of assisting process inputs needed to process the operand. This allows for description of the exchange of the operand between elements of the system.

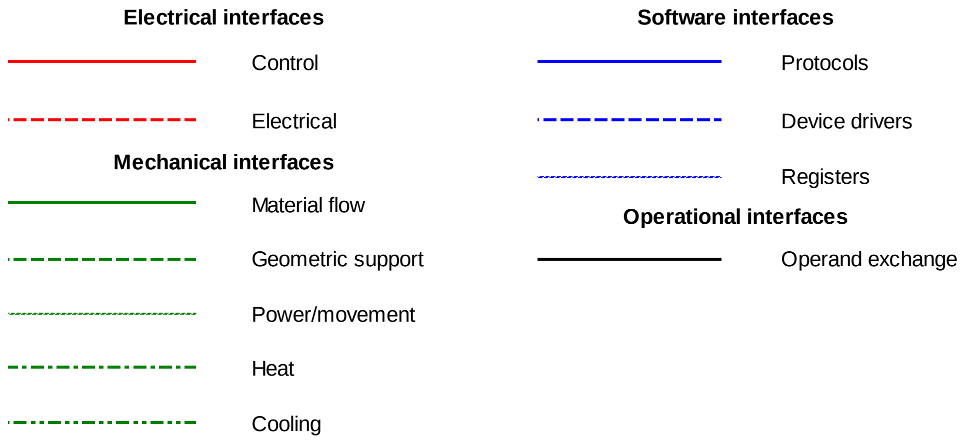


Figure 58 - Types of interfaces in TSV-2

To facilitate effective management of interfaces in the system, the interfaces are defined as belonging to only one of the Key components they connect. This Key component and its responsible stakeholder(s) has the ownership of the interface and must both ensure that the interface is designed to allow integration of the connected Key component or variants of Key components, and control changes to the interface over the life-cycle of the technical system. Part 5 of this dissertation describes a standard for reference designation systems (ISO/IEC 81346-3) that allows for assigning interfaces or interactions a reference designation that will uniquely identify the interface through the Key component that is its master. Such a reference designation can be added to the line representing the interface in the diagram.

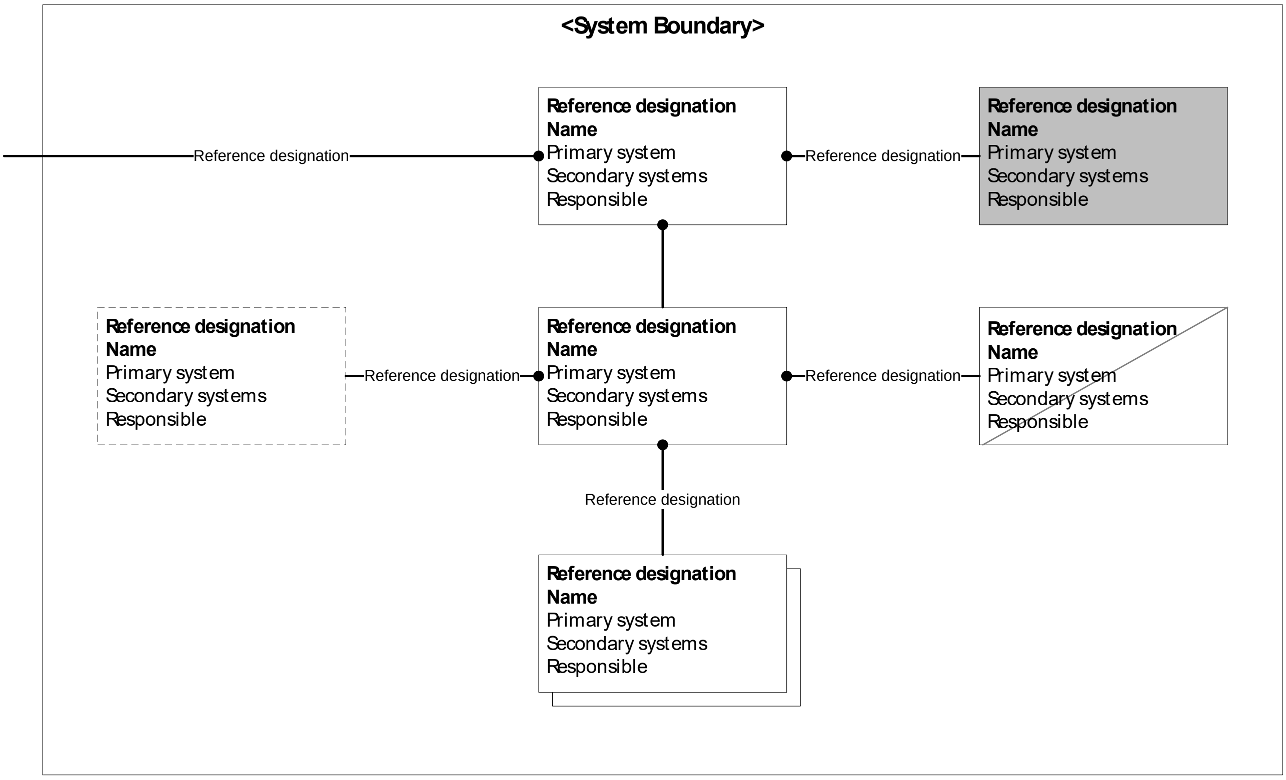


Figure 59 - The syntax of the TSV-2 Interface diagram

The Interface diagram as presented by (Bruun and Mortensen, 2012) models both internal interfaces between Key components, and interfaces to the external environment of the modeled system. In TSV-2 the external environment encompasses both the other operators of the production system (Human System, Information System, Management System and Active & Reactive Environment) as well as the general environment of the production system. Among other things this means that the TSV-2 models the interfaces between the technical system and the Human System.

Describing other structures

The TSV-2 allows stakeholders to define different layers in the model showing other structures in relation to this structure. Secondary layers represent other viewpoints on the structure of the Technical system and can represent structures such as a module structure or CAD structure of the system, as is suggested in the Interface diagram modeling formalism (Bruun and Mortensen, 2012). Other viewpoints on the structure may include different encapsulations of design units e.g. modules. These new encapsulations/groupings of design units are in themselves new design units. Any new groupings/encapsulations of design units shown in such layers of the TSV-2 follow the same syntax as the existing design units in the diagram i.e. they are represented by the same block format, but with the encapsulated design units nested within the blocks. There is no limitation on how many blocks may be nested within each other as for example a module may contain sub-modules. If it is not possible to assign a system relation to the new encapsulations i.e. a module cannot be associated with a specific system, then no system reference is given. The layers representing other structural viewpoints may be shown either as separate diagrams or as filters on the basic structure if the TSV-2 is presented digitally.

Using the TSV-2

There are several ways of using the TSV-2 in the description of the architecture depending on the purpose of the description. If the TSV-2 is used in an architecture activity concerning the design of new production systems, then the TSV-2 can well take the same shape as the FD described by Bruun, where the TSV-2 in the end will primarily describe the part structure of the Technical system and can be used to reason about the part design of the technical system. If the stakeholders instead have a need for mostly applying the system perspective to the Technical system, then the structure of the TSV-2 consists mostly of organs, and can be used to reason about the functional design of the Technical system and the interaction of functional systems to achieve the desired functionality.

Separating the system structure and part structure

If the architecting stakeholders feel the need to maintain a model of the organ structure while also developing the diagram to model the part structure of the Technical system, it could make sense to maintain two separate diagrams. The TSV-2 allows the architecting stakeholders not to model the organ structure and the part structure in the same diagram, and instead keep the two structures separate, while still allowing for mapping from the functional system structure to the part structure. This split can be achieved by dividing the TSV-2 into two diagrams instead of one i.e. the Generic organ diagram presented by (Harlou, 2006) and a Generic part diagram (see Figure 60). The two diagrams would still follow the syntax of TSV-2. The Generic Organ Diagram would only model the structure of the functional sub-systems i.e. the Organ structure, and the generic Part diagram would only model the part structure of the Technical system. References for the affiliation between parts and functional sub-systems would still be included in the Generic Part Diagram, and would therefore still enable mapping between the functional sub-systems and the parts. The division of structures would simply allow stakeholders to treat the modeling of the functional sub-systems and the parts separately. Together with the PCV-2 the architecture description would then include a Generic process diagram (or Generic activity diagram), Generic organ diagram and a Generic part diagram, to describe the Activity system, Organ system and Part system respectively.

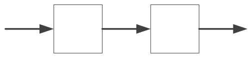
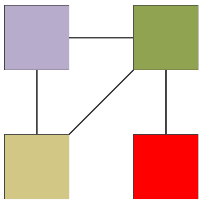
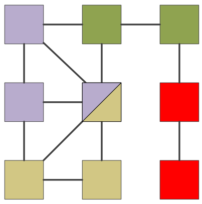
PSAF diagram	PCV-2 Process flow diagram	TSV-2 Interface diagram	
	(Generic process diagram)	TSV-2.1 Generic organ diagram	TSV-2.2 Generic part diagram
Structure	Activity structure 	Organ structure 	Part structure 
Elements	Technical activities (Processes)	Organs (Functional systems)	Parts
Relations	Interactions	Interactions	Interactions + Interfaces

Figure 60 - Dividing the TSV-2 into two diagrams, a Generic organ diagram and a Generic part diagram.

Simplified application of TSV-2

Depending on the descriptive needs of the stakeholders the TSV-2 can be generated in both detailed and non-detailed forms with omission of some of the information associated with the standard designs and interfaces. This could for example be the case if the Technical system is very simply by design, or if there are very few stakeholders involved and the communication and documentation need is low. An example of TSV-2 from the development of a platform for a rotary indexing table is shown in Figure 61. The model is reduced to the main parts of the system and consists of only a single layer (the basic structure of parts). Responsibility for the parts and system affiliations has not been assigned and the reference designation system has not been applied. The model was applied in the early stages of a platform development project to gain an overview of the main design units of the system, to establish which units were optional, existing and standard.

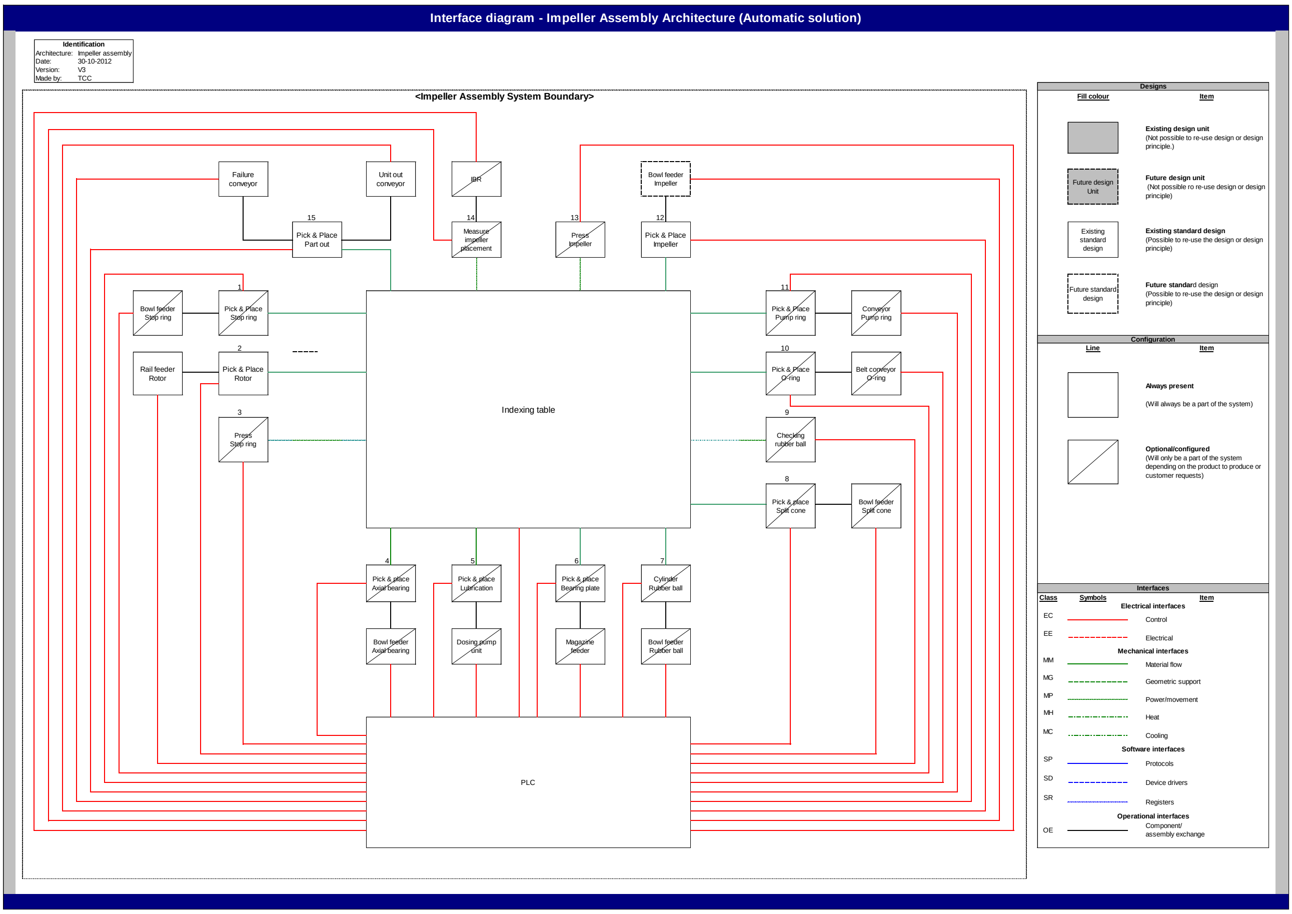


Figure 61 - Example of simplified TSV-2

Remarks on development

The usefulness of the Interface diagram and its predecessor the Generic Organ Diagram has been proven already by other researchers. Based on testing of the modeling formalism as part of the description of architecture in the Chinese department of the production system development, it can be concluded that the TSV-2 is relevant to include as a model in the description of the technical system. The modeling formalism was seen to provide a useful description of the basic system structure of the production system in question, and could be used to facilitate a discussion of how the system is configured, which Key component should be standard, and what Key components had yet to be developed. Due to the limited complexity of the modeled system, the IDF was only made with a single layer and information regarding system affiliation and responsibility of the design units was omitted. It was found that it was a bit difficult to explain the modeling formalism to the Chinese engineers. If this was due to short comings on the part of the researcher or due to languages difficulties (both researcher and subjects were communicating in English, a second language for both), that the approach was new to the engineers, is difficult to say. It was shown that it was possible to describe the system in the TSV-2, and additionally that it was possible to model the changeability of the system.

Applying a functional system viewpoint to the description of the technical system works well to ensure the functionality of the production technical system in relation to the required technical process to be carried out, but the functional system viewpoint is not ideal for the build, sourcing, maintenance, or other purpose. These purposes need a different view on the system. Regardless of what other views are included in TSV-2 besides the basic functional structure, the TSV-2 must at minimum include the basic system structure of the Technical system. Any other structures shown in separate diagrams or applied as filters on the system diagram are optional for the architecting stakeholders. It is also possible to apply a simplified version of the TSV-2 where information such as responsibility and system affiliation for the Key components is not specified, but at minimum it should be possible to uniquely identify the Key components.

24.3 TSV-3 Configuration diagram

The TSV-3, Configuration diagram, addresses concerns regarding the physical configuration of the Technical system i.e. the physical arrangement of parts in the system.

What does it do?

Where the TSV-2 can show the qualitative (i.e. functional) structure of the Technical system, the TSV-3 offers a description of the quantitative (physical) structure of the Technical system showing the physical configuration of parts. The model describes the physical layout of the technical system and the differences in the layout resulting from the changeability of the system. The diagram also describes the preferred (or dictated) progression of changes in the system configuration to obtain specific production capacities or functionalities.

What is it?

The TSV-3 is a simple 2D representation of the principle physical structure of the technical system that shows where in the layout the variants of the parts of the Technical system can be configured. The diagram is supplemented by specific layout configurations showing the specified or preferred configurations of parts and an overview of the classes of parts that can be configured in the system e.g. if there are many different variants of parts (e.g. equipment) that can be used in the system to achieve the same functionality.

Intended usage

The intended uses of the TSV-3 include:

- Description of the basic principle layout of the Technical system
- Specification of preferred principles of changeability in the Technical systems

Introduction to the modeling formalism

The focus of the TSV-3 is on the physical layout of the Technical systems and the possibilities for configuring the layout to suit the specified functionality of the system. The TSV-3 describes the basic physical structure of the Technical system and shows where parts (variants and optional parts) can be configured. The model also describes the preferred configuration principles in relation to the basic physical structure, as determined by the system stakeholders. These preferred configurations constitute a subset of all possible configurations, and document the configurations that are judged by stakeholders to provide the best design of the technical system to realize different production capabilities as specified in the PCV.

Data/information in model

The TSV-3 includes information on the layout of the technical system, the variety of the elements that can be configured in the layout, and information regarding the configuration principles to be followed when changing the system or when designing variants of systems in a family of Production systems.

Type of model

The TSV-3 is a schematic diagram that uses abstract shapes to represent the actual physical parts of the system and shows the spatial configuration of the parts relative to each other.

Modeling formalism

The TSV-3 models the physical configuration of the parts of the Technical system. The model has its basis in the modeling of quantitative structure presented by (Tjalve, 2003). Where (Tjalve, 2003) describes the process of variation in the quantitative structure of a product, the TSV-3 describes the specific configuration of a system (the Technical system) and the preferred variations in the configuration. There are two elements to the TSV-3 (:

1) A layout diagram showing the basic quantitative structure of the Technical system. The modeled elements of the system correspond to the same parts of the Technical system that are modeled in the TSV-2. If more than one principle layout is possible within the production system architecture, then the diagram contains a corresponding number of principle layouts.
2) Diagram of preferred configurations: Diagrams showing the preferred principles of change in the configurations of the system to be followed when changing the system configuration to generate variants of the system or when changing the design of one system to achieve different production capabilities specified in the PCV.

Layout diagram

The layout diagram of the TSV-3 uses geometric shapes to represent the classes of parts of the Technical system. Variants of the parts belonging to the same class may exist. The shapes can either be simple rectangles or they can approximate the cross-sectional geometry of the part class they represent. Each shape in the diagram is given the name of the corresponding class of parts. The name is the same as the one used in the TSV-2 to represent the part. Each part may also be given the same reference designation used in TSV-2 to allow mapping between the two model kinds, or a simple numbering scheme may be used for referencing only within the model (see Figure 62)

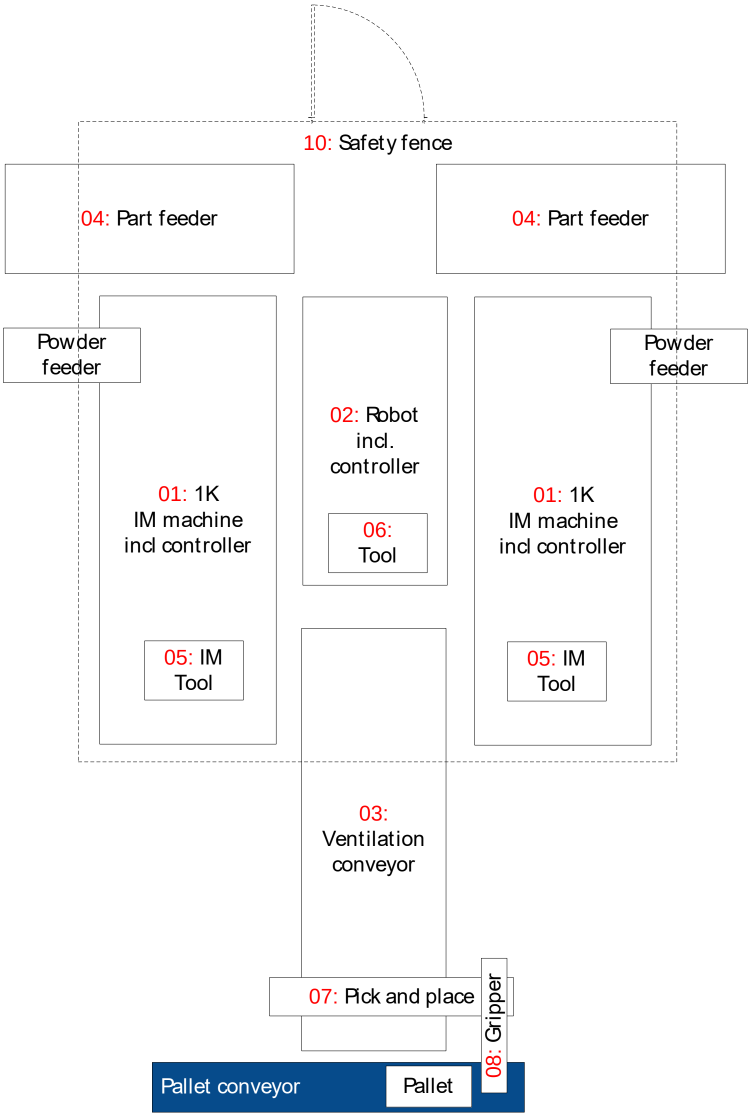
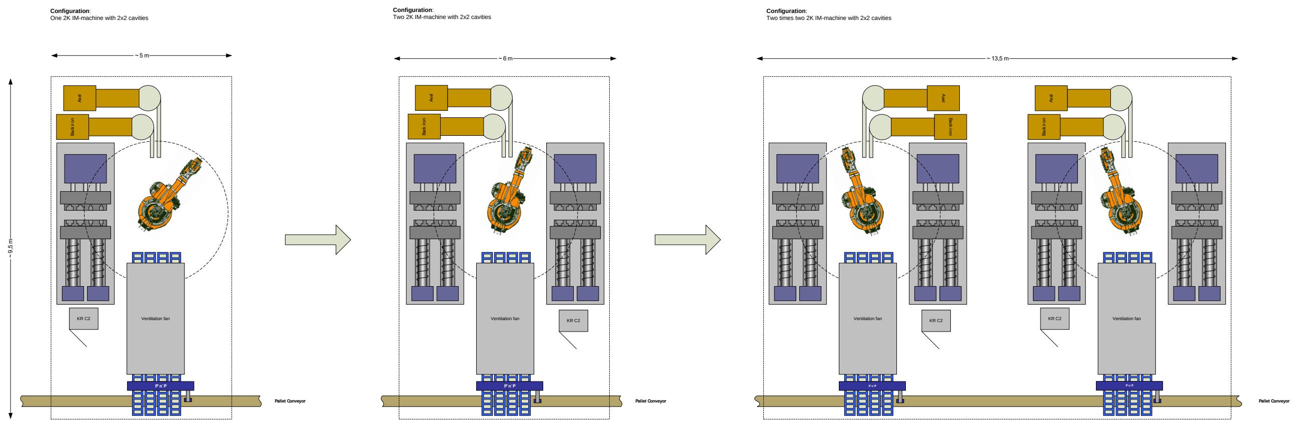


Figure 62 - Layout diagram in TSV-3

Preferred configurations diagram

The diagram shows the preferred configurations that can be generated based on the layout diagram. The diagram should explain the main principles of configuration used within the technical system to address the capability requirements defined in the PCV e.g. scaling of capacity or addition of new capacity. A list of configured parts can be included for each configuration, along with key relevant information regarding the capability of the system e.g. estimated production capacity, investment level, floor space needed, operand archetypes that can be produced, etc.



Process time	40 sec
Floor space	47 m2
Max capacity	~ 750.000 pcs/ year
Cycle time	20 sec/ part
Price	?

Equipment config
- 1 Six axis robot (KR 16-2)
- 1 Robot tool
- 2 IK IM machine
- 2 IMM mould (4 cavities)
- 1 Component feeder
- 1 Pick & Place unit w. tool
- 1 Ventilation conveyor
- 1 Cell controller
- 1 Safety fence

Robot sequence
- Pick up 4 axels
- Move to IM machine 1
- Take out 4 rotor packs
- Insert components
- Move away
- Place rotor packs on ventilation conveyor

- Pick up 4 axels
- Move to IM machine 2
- ...

Process time	40 sec
Floor space	57 m2
Max capacity	~ 1.500.000 pcs/ year
Cycle time	10 sec/ part
Price	?

Equipment config
- 1 Six axis robot (KR 16-2)
- 1 Robot tool
- 2 IK IM machine
- 1 IM mould (4 cavities)
- 1 IMM mould (4 cavities)
- 2 Component feeders
- 1 Pick & Place unit w. tool
- 1 Ventilation conveyor
- 1 Cell controller
- 1 Safety fence

Robot sequence
- Pick up 4 back irons and 4 axels
- Move to IM machine 1
- Take out 4 axel assemblies
- Insert back irons and axels
- Move to IM machine 2
- Take out 4 rotor packs
- Insert axel assemblies
- Move away
- Place rotor packs on ventilation conveyor

Process time	40 sec
Floor space	~ 125 m2
Max capacity	~ 3.000.000 pcs/ year
Cycle time	5 sec/ part
Price	?

Equipment config
- 1 Six axis robot (KR 16-2)
- 1 Robot tool
- 2 IK IM machine
- 1 IM mould (4 cavities)
- 1 IMM mould (4 cavities)
- 2 Component feeders
- 1 Pick & Place unit w. tool
- 1 Ventilation conveyor
- 1 Cell controller
- 1 Safety fence

Robot sequence
- Pick up 4 back irons and 4 axels
- Move to IM machine 1
- Take out 4 axel assemblies
- Insert back irons and axels
- Move to IM machine 2
- Take out 4 rotor packs
- Insert axel assemblies
- Move away
- Place rotor packs on ventilation conveyor

Figure 63 - Diagram of preferred configurations in TSV-3

Example

Figure 64 shows an example of the TSV-3 for an Injection molding cell that is a part of an architecture for a family of production lines. The Example consists of two principle layouts, and has a number of possible configuration options displayed in the second part of the TSV-3. The configuration principles are defined to fit the production capacity scaling of the production system and are divided by operand archetypes (as defined in the PCV-1). The preferred configurations/or specified principles of changeability in the system configuration are the primary scaling principles followed in the configuration of the system, however it is possible to generate many other configurations of the system. Using the TSV-3 it is possible to not only describe the configuration options of a family of Technical systems, it is also possible to describe the configuration steps in throughout the life-cycle of a single technical system, and to associate this with the Change impact roadmap and the Production capacity plan to describe how the impacting technology and products and the changes in the production mix and capacity of the system are connected to the changes in the configuration of the system.

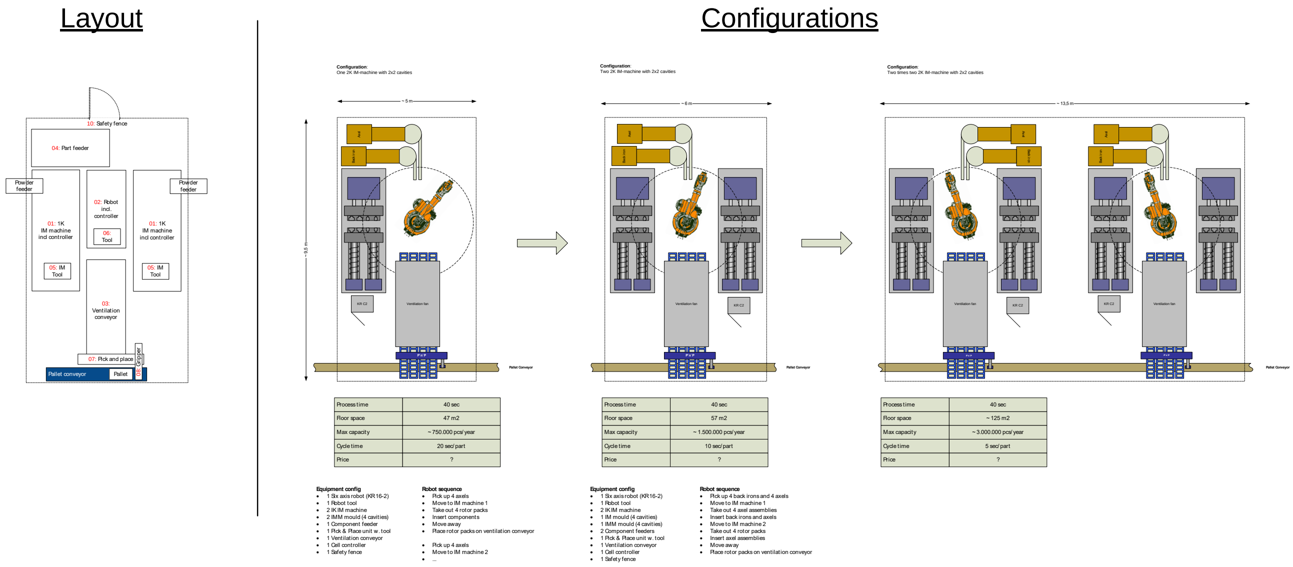


Figure 64 - Example of TSV-3 Configuration diagram for an injection molding cell. Three main configuration steps are used to scale capacity in the production system.

Remarks on development

The TSV-3 has been tested among other for a larger family of production systems covering a key sub-component in the products of Grundfos. The model has shown some usefulness in describing the configuration of the Technical system, but the model is limited in its expression because of the 2D nature of the diagram and the limitations in showing the 3D layout of the system elements. Multiple diagrams are needed to show the different cross-sections of the system, if the layout changes in all three axis. There is also a limitation in the possibility of modeling small elements of the system that may be mounted inside other elements. As such it is believed that the TSV-3 should only be used to show the configuration of the larger elements of the system, and not be used to show small scale detailed configurations. The diagram also has limitations in showing configurations when there is a large difference in the physical size of elements configured to be positioned in the same physical location. The diagram in this case should be focused more on the relative physical location of the configured elements, and less on the dimensional relativity of the modeled elements.

24.4 TSV-4 Technical System Master Plan

The TSV-4, Technical System Master Plan, provides stakeholders with an overview of the variety of the Technical system as it is perceived by different stakeholders. The TSV-4 enables the stakeholders posing application requirements to the Production system and the stakeholders responsible for the engineering design and build of the Technical system, to enter into a discussion on system variety from their respective viewpoints. This discussion covers the required system variety and the means by which that variety is met by the system design. The TSV-4 also serves a basic documentation of the main application related requirements for the Technical system, the organ composition, and the part composition of the Technical system. The TSV-4 also enables analysis of the variety with the purpose of changing the variety to achieve an increase or decrease in variety as perceived by the different stakeholders.

In general the TSV-4 can be said to provide stakeholders with a means of linking between the application layer of the architecture and the design layer of the architecture with regards to the Technical system, and to enter into a discussion on the design variety necessary to address the different applications of the production system within the company.

What does it do?

The TSV-4 models the structure and variety of the Technical system as perceived by different stakeholders in the architecting process in order to facilitate joint decision making regarding the structure and variety of the Technical system. The TSV-4 expresses the variety through modeling of the composition and attributes of the elements of the Technical system, the other elements of the production system and the general environment of the production system. The TSV-4 models both the total space of variety within the architecture, as well as the variety of specific Technical systems and archetypes of Technical systems.

What is it?

The TSV-4 is an object oriented modeling formalism that represents an application of the Product Family Master Plan modeling formalism (PFMP) to the domain of Production system design instead of Product design. The TSV-4 frames the PFMP modeling formalism in the context of production system architecture, and it adds slight modifications to the syntax of the PFMP and as well as the documentation of specific variants of technical systems and archetypes of technical systems.

Intended usage

The TSV-4 can be used in a design capacity to analyze the variety of the Technical system to ensure that the specified production capability and other application related requirements and constraints are suitably addressed in the design of the technical system. Additionally the TSV-4 can be used to document the variety of specific technical systems derived from the architecture, and to define archetypes of systems as references to be used by the organization. The intended usage of the TSV-4 includes:

- Analysis, documentation and communication of the variety of the Technical system as perceived by key stakeholders
- Discussion of design phase out and phase in
- Definition of new variety to develop
- Standardization of design
- Definition of system archetypes
- Definition of platforms

Introduction to the modeling formalism

Both when a technical system or family of systems is initially designed and later on changed throughout its life-cycle, there is a need to establish an overview of the structure and variety of the Technical system (family) from the perspective of the stakeholders involved, to support decisions making regarding the existence and design of system variants. Such an overview must both document and communicate the structure and variety of the system (family) in order to support analysis and synthesis of the design.

According to (Andreasen, Hansen and Mortensen, 1996) several structures can be described within a product or product family depending on the applied viewpoint, and the same is true of production systems. The TSV-4 allows stakeholders to get an overview of the structure and variety of the Technical systems from the viewpoint of two main groups of stakeholders:

1. Stakeholders that formulate the application related requirements and constraints that govern the design of the Technical system
2. Stakeholders responsible for the synthesis of the Technical system

The stakeholders belonging to group one are also referred to as the Customers of the production system. This group may include stakeholders from many parts of the organization incl. but not limited to Production, Finance, Purchasing, Product development, etc. as these are all customer for the production system in one way or another. The stakeholders belonging to group two are the stakeholders responsible for the engineering design of the technical system or the build of the system.

The viewpoint of stakeholders belonging to group one represents a kind of application requirements viewpoint in relation to the architecture phenomenon, meaning that the variety of the Technical system is viewed in relation to the requirements posed by the stakeholders to the various applications within the company that the production system is intended to be used in. The application requirements may be posed as both behavioral and structural requirements, and the variety perceived by the stakeholders will consequently also be viewed in terms of the Technical system's behavioral or structural characteristics. This view of the Technical system variety is represented by the Customer view in the PFMP modeling formalism and the TSV-4.

The viewpoint of stakeholders belonging to group two represents an engineering design viewpoint and is focused on the constituent design of the production system. Two views of the constituent design are included in the description of the variety within the TSV-4, an organ view and a part view representing the functional and physical aspects of the system variety. The variety expressed in these two views is the variety of the organ system and part system respectively. The organ view is represented by the Engineering view and the part view is represented by the Part view of the PFMP modeling formalism.

In total the combined viewpoints of the two stakeholder groups can be said to provide a link between the application layer and the design layer of the architecture phenomenon, by allowing stakeholders to map between variety as seen by the stakeholders formulating the application requirements and the stakeholders responsible for the constituent design of the system.

Views in the model

The TSV-4 represents quite a direct application of the Product Family Master Plan modeling formalism (Harlou, 2006) to the description of technical systems in production systems instead of products. Slight changes are made to the modeling formalism's syntax and semantics to fit the domain, but the main elements remain the same. Among the changes are the variant table and changes to the modeling syntax that were introduced in PCV-1. The part view and engineering view are also seen as views belonging to the same group of stakeholders related responsible for the system design, and is not attributed to the engineering and production as is the case in the PFMP.

The TSV-4 is an object oriented modeling formalism that models the variety of the Technical system in three views i.e. the Customer view, Engineering view and Part view as object structures with mapping to the instances of specific Technical systems (see Figure 65). The three views are explained as follows:

Customer view: The customer view describes the variety of the Technical system from the perspective of the stakeholders posing the application related requirements to the production system. The Customer viewpoint represents a summation of the key capability and design requirements to the production system expressed in both the PCV and TSV. These requirements may relate to both the functional behavioral and structural characteristics of the technical system, and the variety may be expressed either directly as variety of the Technical system's constituent elements, or it may be expressed indirectly by the Technical system's surroundings in the model of a transformation system.

Engineering view: The engineering view describes the Technical system from a functional viewpoint and should seek to explain how the functional elements of the Technical system vary from the viewpoint of the stakeholders responsible for the functionality of the Technical system. The view describes the variety of the organs of the system.

Part view: The part view describes the Technical system from a physical viewpoint and should seek to explain how the physical elements of the Technical system vary from the viewpoint of the stakeholders responsible for the physical design of the Technical system. The part view describes the variety of the part system in the technical system i.e. the physical structure of parts in the Technical system.

Because of the explained meaning of the three views, it could be considered if the views should be renamed to something more appropriate in the TSV-4 e.g. the Application view, System (or organ) view and Part view, however this is not done.

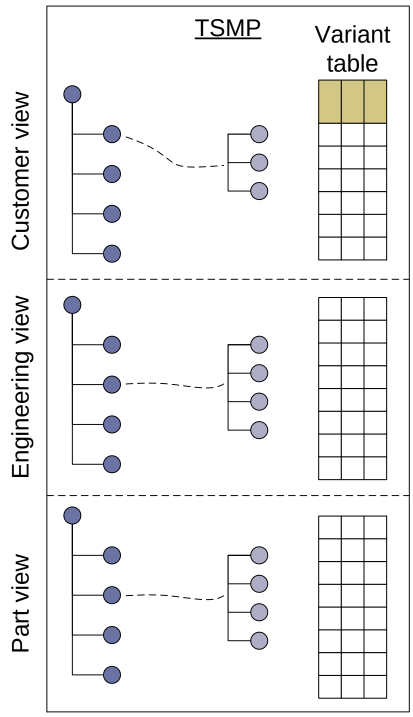


Figure 65 - Overview of Technical System Master Plan.

Structure of the model

The TSV-4 applies the same concepts of object-oriented modeling that are applied in the PCV-1 but models the Technical system instead of the input and output operands of the system. Just as the PCV-1, the TSV-4 consists of three main elements i.e. a Part-of structure, Kind-of structure and Variant table. The difference is that the modeled entities are elements of the technical system and not the processed operands. The syntax used in the two structures and the variant table is therefore the same as that of the PCV-1 (c.f. section 23.1).

Modeling in the customer view

The customer view is intended to provide a description of the variety of the technical system as it is perceived by the stakeholders formulating the application requirements to the production system. These stakeholders may belong to many different parts of the company, and are not limited to the utilization stage of the production system life-cycle. The stakeholders may just as well belong to the company functions associated with financing, purchasing, technology development, Human resources, logistics, etc. because the production system may have a role to play in any of their activities, just as it will have a role to play in the production where it performs a specific production task. The interests of these stakeholders relate to both the structural and functional behavioral and structural aspects of the production system, and their view of the variety of the technical system will therefore similarly be seen in these terms. Additionally the variety of the technical system is not necessarily perceived in terms of the direct variety of the system's functional and structural characteristics. The variety can also be perceived indirectly in terms of the variety of the surroundings or contexts in which the system can enter. That is to say, that the system variety can be described both directly in terms of the variety of the system's constituent design and functionality and indirectly by way of the variety of the elements to which it has a relation. Four ways of modeling the variety are therefore applied in the Customer view:

- Feature modeling: Variety of functionality or structural composition
- Interface modeling: Variety of the interfaces to the system's surroundings
- Technical process modeling: Variety of the elements of the technical process system
- Environmental modeling: Variety of the general environment in which the system is expected to be used.

Note that feature modeling, interface modeling and technical process monitoring are included in the PFMP modeling formalism. The TSV-4 makes some changes to the technical process modeling of the PFMP, and adds the environmental modeling as a way of modeling variety in the customer view.

Feature modeling in the customer view

Feature modeling is presented in the PFMP modeling formalism as a modeling of the variety of functional or structural aspects of a product that are seen as the distinguishing or positioning properties in the market. In the TSV-4 this is translated as variety of the functionality or physical parts of the technical system that are seen as distinguishing characteristics. Examples of this could be the inclusion of real-time production line monitoring equipment, performance displays, video monitoring equipment, measurement equipment, etc. (see Figure 66).

Interface modeling in the customer view

Interface modeling is presented in the PFMP modeling formalism as a description of the variety of physical interfaces to the customer's application environment in which a product is used. For production systems the customer's application environment will typically cover the utilization stage of the production system's life-cycle and the interfaces are either to the technical systems of other production systems e.g. because the production system is a part of a larger production system (in the way that a production cell might be a part of a production line), or they are interfaces to the technical systems of the factory itself e.g. the supplies in the factory or the floor of the factory. Interface modeling for the technical system therefore describes the perceived variety of interfaces to the other operators of the production system, or to other production systems or to the factory in which the technical system is located. The interfaces could include electricity, water, air, chemicals, data, or man/machine interfaces (see Figure 66).

Technical process modeling in the customer view

Technical process modeling is presented in the PFMP modeling formalism as a description of the perceived variety of a product in regards to the technical process system in which a product is used. And this technical process system is seen as a descriptor of the applications in which the product is used. The TSV-4 also applies this perspective to application modeling, but as explained in the definition of the architecture phenomenon, does not limit the applications of the production system to the technical process that must be achieved in the utilization stage of the life-cycle. The production system, and the technical system by extension, has many different applications. The TSV-4 therefore only describes the variety in terms of the applications found in the utilization of the technical system in the performance of a technical process during production.

The PFMP modeling formalism applies a model of the technical process system in which the system is described by four elements: the technical system, human system, environmental system, and the technical process. The TSV-2 modifies this to follow the model of the technical process described in section 5.1, and thus describes the variety of the technical process system as the variety of the following elements, examples of which can be seen in Figure 66:

- Operators
    - Technical system
    - Human system
    - Information system
    - Management system
    - Active & reactive environment (i.e. environmental system)
- Technical process
- Operands

Environmental modeling in the customer view

As an addition to the PFMP modeling formalism, the TSV-4 also describes the variety of the technical system indirectly through the variety of the General environment in which the production system (and the technical system by extension) is to be applied. The general environment represents a description of the customer’s applications of the production system that do not necessarily have to do with the technical process as covered by variety description through technical process modeling. The general environment represents a greater cross section of the applications in which the production system must play a part. Modeling of variety in relation to the general environment pertains to the following environments on a regional, national or international level:

- Physical & chemical
- Geographic
- Organizational
- Economic
- Legal
- Political
- Cultural
- Ideological

Particularly interesting among these are the physical & chemical, geographic and legal environments. Some examples for perceived variety could be the variety of the following, examples of which can be seen in Figure 66:

- Physical & chemical
    - Humidity
    - Temperature
- Geographic
    - Specific countries
    - Specific production locations
    - Language zones
    - Altitude zones
    - Climate regions
    - Transport options (to production location)
- Legal
    - Export restrictions on production technology
    - Import restrictions on production equipment e.g. as a consequence of protectionism for local equipment suppliers
    - National or regional safety regulations
    - Patent restrictions

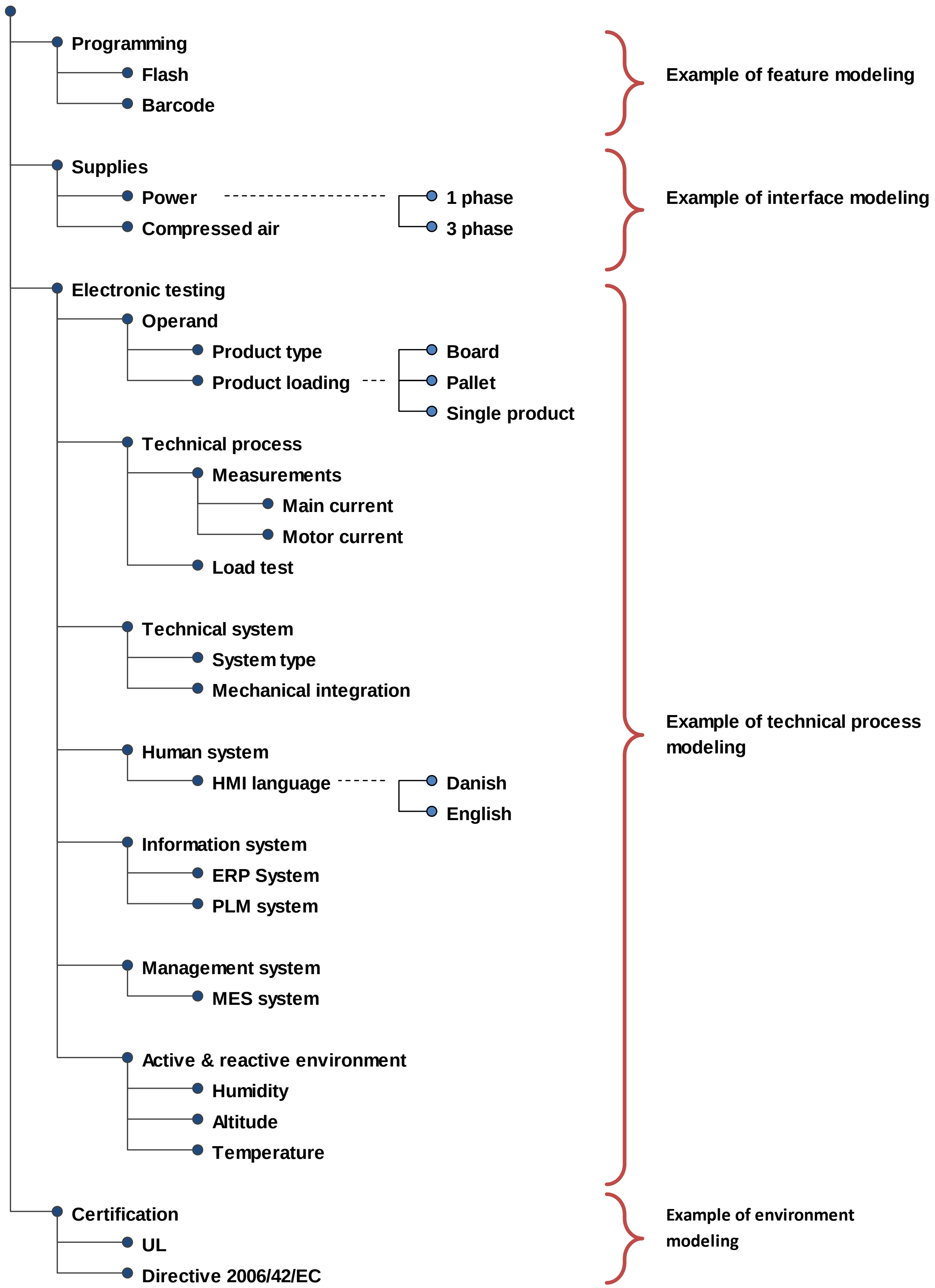


Figure 66 - Example of the four ways of modeling in the customer view.

Modeling in the Engineering view

The purpose of the engineering view is to describe both the functional design of the technical system and the variety thereof as seen by the stakeholders responsible for the functional design of the technical system. The view models the structure and variety of the functional elements of the technical system, and the view thus presents both the variety of the functional elements and provides stakeholders with an overview of the system's functionality as represented by the functional elements. As previously explained, the functional elements of the technical system are also referred to as organs, and the structure of functional elements is the organ structure of the system. The engineering view therefore applies a single modeling method i.e. organ modeling.

Organ modeling in the Engineering view

The engineering view models the organ system of the technical system, and the classes in the view therefore represent organs. The part-of structure in the TSV-4 models the generic organ structure, which is equal to the generic organ structure of the TSV-2 Interface diagram. The variants of the organs are modeled in the kind-of structure and these are equal to the variants that are indicated for classes of organs in the TSV-2 Interface diagram.

Just as in the TSV-2 Interface diagram, organs may consist of other organs. (Hubka and Eder, 1988) suggest referring to different levels of organs as organisms, organs or partial organs, to represent the different levels of organ groupings. For the purpose of modeling in the organ view it is sufficient to think of the different levels of organs simply as super-organs and sub-organs rather than to assign specific names to different levels of organ groupings.

Modeling in the part view

The purpose of the part view is to describe the part design of the technical system and the variety thereof as seen by stakeholders responsible for the part design of the technical system. The part view applies a single method for modeling the part system based on a viewpoint chosen by the architecting stakeholder. This viewpoint may relate to a specific life-cycle stage or a certain system breakdown. In many cases the structure may be equal to a generic BOM.

Part modeling in the Part view

The part view models the part system of the Technical system, and the classes in the model therefore represent parts. The part structure in the TSV-4 models the generic part structure, which is equal to the generic part structure of the TSV-2 diagram. The variants of parts are modeled in the kind-of structure and these are equal to the variants of parts that are indicated for classes in the TSV-2 Interface diagram.

Modeling in the variant table

The variant table in the TSV-4 models the instances of Technical systems and archetypes of Technical systems. On the whole the table is the same as in the PSV-1 with some slight alterations to the column headers that identify the instances of technical systems and archetypes. Rather than status, the archetypes of technical systems are given an example system i.e. a reference to a specific technical system that is exemplary of the archetype. The changes can be seen in Figure 67.

	<Status> <ID> <System name>	<Status> <ID> <System name>	<Status> <ID> <System name>	<Status> <ID> <System name>	<Status> <ID> <System name>	<Example system> <ID> <Archetype name>	<Example system> <ID> <Archetype name>	
Super-part A	1	1	1	1		1		← Cardinality
Attribute 1	F	F	N	N		N		← Attribute value
Attribute 2	35	50	80	90		<95		
Super-part A:Sub-kind 1	1	1						
Super-part A:Sub-kind 2			1			X		
Super-part A:Sub-kind 3				2		X		

Figure 67 - TSV-4 variant table format

Example

Figure 68 shows an example of the TSV-4 used to model the variety of a family of Electronics testers. The model kind was tested to see if it was applicable to testing of a family of production systems with a large degree of variety. The model only includes the Customer view and Engineering view, since the Part view was deemed unnecessary in the project. The mapping in the variant table was performed for systems that exemplified the system archetypes within the production system family, to avoid mapping all systems. Based on this model it was possible to perform analysis of the similarities/sharing between systems. The model also allowed for stakeholders to enter into a discussion of what system archetypes should be phased out and which should form the basis of future development. It was found that the model afforded stakeholders a condensed overview of the variety within a large family of systems, which was not available through other means.

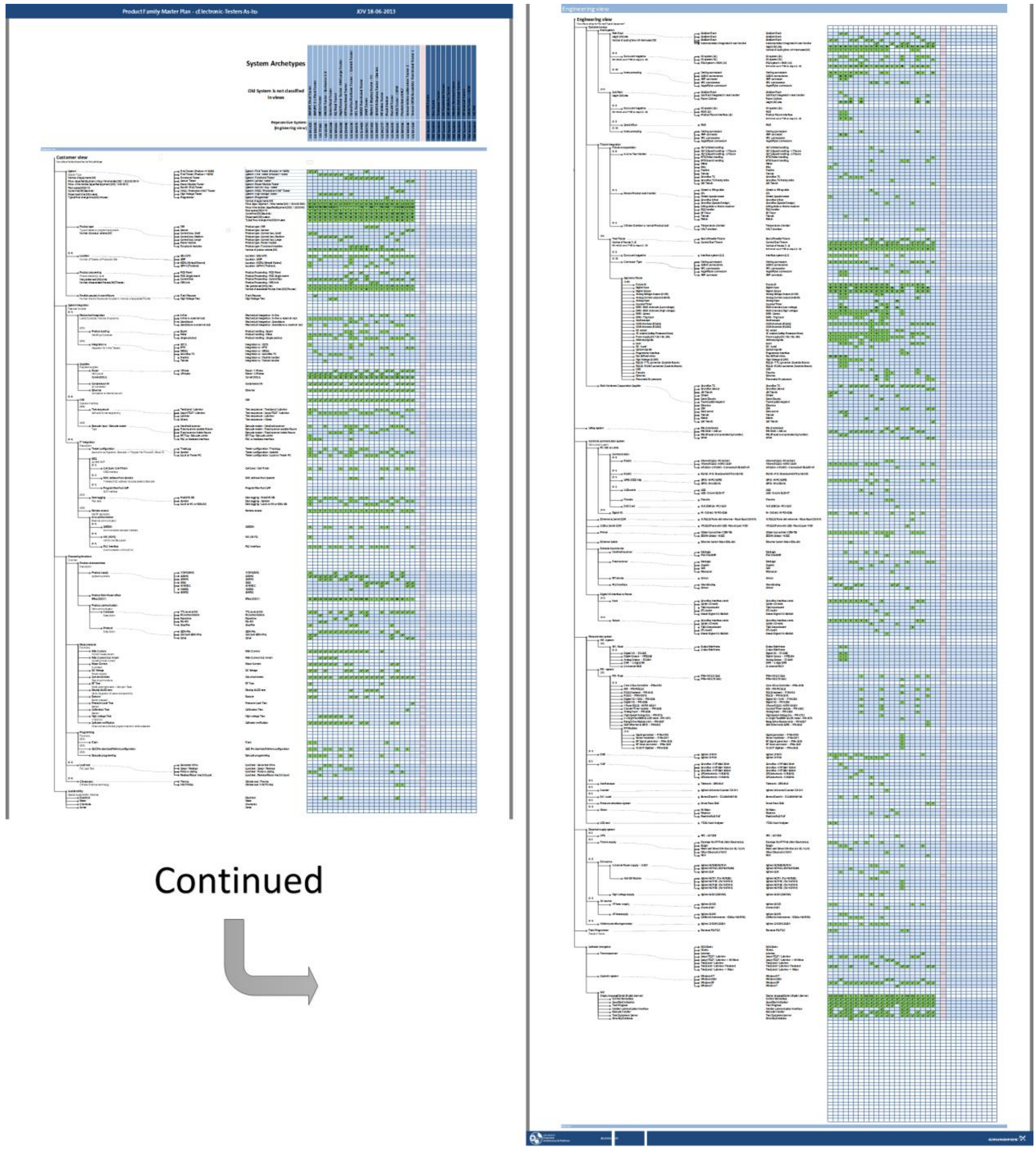


Figure 68 - Example of Technical System Master Plan for a family of Electronics testers.

Remarks on development

The TSV-4 serves as a description of the variety of the Technical system, and can be used not only as a documentation tool, but as a point of discussion and agreement between the architecture stakeholders. Through modeling of the current (as-is) variety and a potential future (to-be) variety, the TSV-4 can facilitate discussion of the future variety of the technical system including which solutions should be phased out or phased in as part of the architecture. The TSV-4 has been used in this capacity within the case company (see example in Figure 69). Testing of the TSV-4 has shown that it is possible for the stakeholders in the case-company to adopt the TSV-4 for variety modeling with only minimal introduction to the modeling formalism and guidance in the modeling process. The additions to the modeling in the customer view compare to the PFMP modeling formalism have been deduced from the observed differences in the customer views of the case models, and the necessity of adding a representative system for archetypes, was deemed necessary to allow the stakeholders a mental reference both when describing the variety and when using the model to discuss good and bad variety and the future development of the architecture.

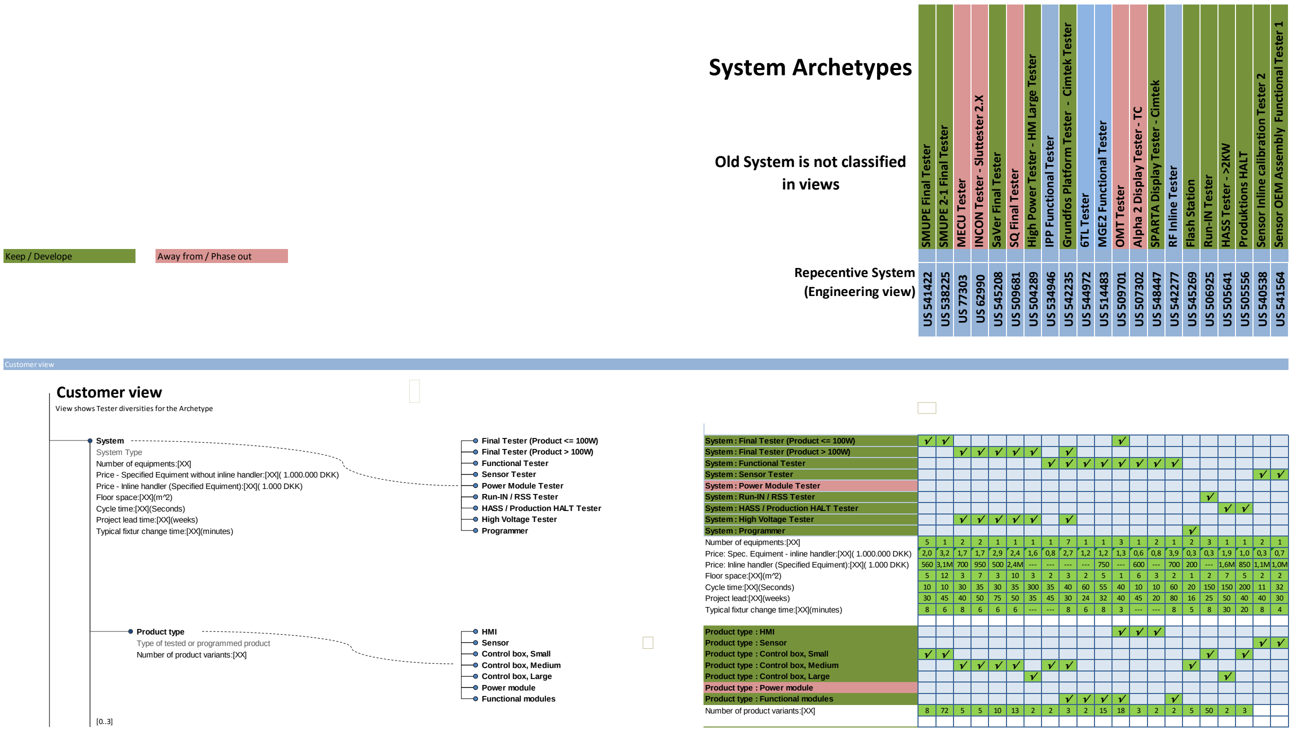


Figure 69 - Modeling of a family of electronic testers. The green and red colors in the model indicate decisions on phase-in/keeping (green) and phase-out/abandonment (red) of variety.

25 Development of viewpoints and model kinds

The viewpoints and model kinds presented in this Part of the dissertation represent the consolidated results from multiple individual projects carried out at the Grundfos Technology Center throughout a period of 2 years and 4 months. This research has both concerned the fundamental nature of the architecture phenomenon for production systems and the means and ways by which architecture can be described, developed and managed. Development & testing of the model kinds have occurred through an action research approach where the researcher has actively engaged in the work of the company as a member of different project teams and as one of the main drivers of a strategic initiative for introduction of an architecture and platform focused development doctrine for production systems and production technologies. The degree of my participation in projects within the company has spanned from being merely an observer to an active participant in the projects' execution. The types of projects I have participated in include:

- Planning & management
- Equipment procurement

- Production system development
- Equipment standardization
- Architecture definition
- Architecture mapping
- Work process development
- Tool & methods development
- Knowledge & competence mapping
- Organizational communication

Some of these projects have been very small while others have spanned a significant portion of the research period. It is no exaggeration to say that the viewpoints and model kinds presented in this dissertation are shaped by the totality of my participation in these projects. Five projects in particular form the basis of the presented viewpoints and model kinds. Their relevance for the developed model kinds can be seen in Table 6 below:

Table 6 - Key projects and their relevance for the developed model kinds.

Architecture level	Production system	Project purpose	PCV models	TSV models
Assortment	Rotor assembly line	Architecture & platform development	PCV-1 PCV-2 PCV-4 PCV-5	TSV-1 TSV-2 TSV-3
Assortment	Electronics tester cell/machine	Standardization		TSV-4
Family	Injection molding cell	Architecture & platform development	PCV-2	TSV-1 TSV-2 TSV-3
Family	Impeller assembly cell	Architecture & platform development	PCV-2	TSV-2
Single system	Rotor assembly line	Procurement project	PCV-2 PCV-3	

The projects cover architecture development/definition, platform development, standardization and procurement, and have provided a broad perspective on the needs for architecture descriptions. In addition to strictly system focused projects, some of the model kinds (in particular the PCV-4 and PCV-5) have been used to model application related information in the mapping of technology domains related to sub-assemblies of the produced products. This has provided additional input for the modeling formalism, and has allowed for testing of the model kinds' broader applicability.

In my participation within the projects I have always strived to seek input for the understanding of the architecture phenomenon, how it should be addressed, and what descriptive tools and methods are necessary to apply an architecture-centric approach to production system design. I have supplemented the research in the projects with the study of other knowledge sources within the company such as study of:

- Work processes & process models
- Organization & collaboration structures
- System designs and descriptions
- Collaboration with suppliers/external partners
- Company strategies, memo's, templates, standards

From the development and testing of the viewpoints and model kinds I can conclude that the range of needs for description of architecture is so varied that applying all of the model kind within a single project would be redundant and provide project stakeholders with unneeded models. The presented viewpoints and model kinds should instead serve as a general set available viewpoints and model kinds, which that afford some flexibility in their use. It should be possible to pick and choose model kinds based on the descriptive need, and it should be possible to apply simplified versions of the model kinds if the descriptive needs are limited. This is the reason that the final contribution of viewpoints and model kinds is said to be represent a consolidated result. This also means that users of the results are meant to select the viewpoints and model kinds that can address their set of architecture related concerns and that simplification of the model kinds are permissible.

26 Conclusion on Part 4

Part 4 has presented a contribution to description of production system architecture, by modeling of aspects of interest for the architecture. The contribution consists of two library viewpoints that are a part of a Production System Architecture Framework suggested in Part 3. The two viewpoints describe the production system architecture in relation to two main categories of stakeholder concerns regarding the architecture, by means of different model kinds. The first viewpoint, the Production capability Viewpoint, helps to frame concerns regarding the production task of the system within the company. This viewpoint covers concerns relating both to the processing capability of the system, but also concerns regarding the systems relation to the technology development of the company. The second viewpoint, the Technical System Viewpoint, helps to frame concerns regarding the design of the technical system within the production system. This viewpoint serves to describe the constituent design of the system, select changeability characteristics, and the variety of the system.

The presented viewpoints and model kinds represent a basis from which stakeholders can select viewpoints and model kinds appropriate for their descriptive needs in regards to their system of interest. The viewpoints and model kinds are consolidated from the development of model kinds in multiple projects within the primary case company, and while they help to frame many architecture related concerns, they do not offer a full description of the production system architecture. Further development of PSAF, including addition of other viewpoints and model kinds, and further testing will be necessary before the suggested reference framework can be said to offer a comprehensive description of the production system architecture in relation to all the different system applications within a company.

Part 5 A contribution to correspondence in a Production System Architecture Framework

Part 5 describes a contribution to information handling for in relation to production system architecture by means of reference Designation System (RDS). The RDS is used as a correspondence kind in architecture descriptions, which allows for correspondence between models of the description. The RDS provides identification and referencing of key design elements and relations within the architecture of a production system. The developed RDS is based on an application of the ISO/IEC 81346 standard series and is part of research into the applicability of the series in architecture-centric design. The RDS addresses the limitations of the standard in unambiguously identifying design elements belonging to different domains; in describing the shared architecture of multiple systems; and in the support for design reuse across multiple production systems.

27 Correspondence rules

In addition to a contribution to description of production system architecture it is the stated objective of this research project to support information handling in regards to the communication of the design of the production system. The need for information handling in the architecting process is quite broad because the architecture is handled in many stakeholder domains, business processes, design tools, documents and IT-systems. PSAF provides the basis for describing production system architecture, and includes a great deal of architecture related information in specified viewpoints. The effective use of the PSAF viewpoints is not only dependent on the use of the individual model kinds, but also very much in the ability to interrelate the model kinds. Relating the model kinds to each other is necessary in order to address more complex concerns, gain a more complete overview of the architecture and to enable communication and understanding between stakeholders. Observations and experience from the case projects have identified the following needs in the use of the different model kinds:

- Viewpoints and model kinds must be interrelated for the purpose of analysis and synthesis of the architecture
- It should be possible to map from the existing viewpoints and model kinds to any new viewpoints or model kinds added to PSAF in the future
- Constituent elements of the production system must be identifiable so they can be managed within the company
- There must be traceability for design elements in the architecture description throughout the production system life-cycle in order to follow the propagation of changes throughout the architecture
- It should be possible to map design elements from the architecture description to documentation and IT tools in the company e.g. equipment drawings and ERP systems.

These needs all relate to the necessity of correspondence between the viewpoints and model kinds, and are focused on the constituent elements of the production system that are the subject of the design process. The contribution to the information handling in this research project therefore covers:

- Documentation of the key constituent elements and relations of production system architecture;
- Identification and referencing of the key constituent elements and relations throughout the production system life cycle and across the documents and IT-systems of the company.

The correspondence rules and correspondence kinds include of PSAF represents the contribution to this research objective. PSAF does not include any other correspondence rules or correspondence kinds to address other need correspondence, since this is outside the scope of this research. Should PSAF be developed further in the future, it would be beneficiary to specify how model correspondence should be handled for all the included model kinds and the architecture description elements included therein.

Correspondence rule for identification and referencing of the production system's constituent elements

Within PSAF it is specified that the main constituent elements of the production system be identified and referenced by means of reference designations that have been assigned to them by a reference designation system (RDS). This includes the technical process, the operators and their relations of type Composition, Attribute and Allocation.

The reference designations assigned to the elements can be used for identification and referencing of the elements and their relations both within the model kinds presented in this dissertation; new model kinds added in the future; other design tools e.g. Design Structure Matrix, Function means trees…, etc.; engineering documentation; and the company's IT-systems. The RDS system constitutes the only correspondence kind included in PSAF.

The RDS provides a generic model that can be used as the basis for referencing the elements and is independent of suppliers, brands, IT-systems, etc.

The RDS developed in this research project is based on the ISO/IEC 81346 standard and research question three represents the necessary research questions for the development of such a system.

28 What is a reference designation system?

In the design of systems (whether they be products or production systems) there is often a need for uniquely identifying a design element. For simple systems or early stages of design, this is often done simply by naming the elements or using brand names, model numbers, serial numbers, inventory numbers, or similar identifiers. Using names as the means of identifying and referencing elements can be very inexpedient, particularly when:

- the system consists of many elements (high complexity)
- the design task is divided between different development organizations
- the name is not sufficiently descriptive to distinguish between elements
- the architecture of the system covers a family of systems
- there can be multiple instances or variants of the element
- there is no central definition/understanding of what is identified/referenced when using the name

Using brand names, serial numbers or similar identifiers can also be problematic, as it locks the architecture to a specific variant or instance of the element. It would be better to not mix the identification of the element to the specific instance or lock the design to a specific chosen implementation (brand, model, etc.). Using this information makes the architecture rigid.

A Reference Designation System (RDS) is one way of addressing the identification and referencing needs in the design of systems and the description of their architecture. An RDS is a system of principles, rules and guides for the formulation of reference designations for objects in systems to be used in the structuring of system elements and their associated information. Reference designations are a type of identifiers for objects defined in relation to the system of which the object is a part. The reference designations can be used to identify the key elements and structures of a system and also provide stakeholders with a common understanding of the constituent elements of a system and their structures. Such a common understanding is important for the collaboration and communication between stakeholders and it allows for mapping between models, processes, design tools, IT-systems and more, where ever the elements of the production system are the subject of an activity within the company. Among some of the intended uses of the RDS developed in this research project can be mentioned:

- Providing an understanding of the models (what are they modeling)
- Mapping between models for referencing and analysis
- Tracking of system elements throughout the production system life-cycle
- Support communication with external stakeholders with a common reference frame
- Provide a more formal means of correspondence not based on “names” for system elements
- Searching and tracing of system elements (for management support)
- Use correspondence that leans on the means of correspondence used in mechanical and electrical documentation (when basing the RDS on ISO/IEC 81346)
- Provide a core reference for the elements and structures of the system
- Provide a means of navigating in complex designs that span domains
- Potentially reuse documentation when codes are standardized
- Allow expression of relations between elements of the same domain and different domain in a format that follows industry practice.
- Interconnect very different models even models that are not currently a part of PSAF e.g. DSM
- Provide a reference for use in both documents and IT systems.
- Link the architecture description to traditional engineering documentation.

Inspiration for PSAF-RDS

The RDS developed in this research project is based on an investigation into the applicability of the standards and technical specification of the ISO/IEC 81346 standard series for architecture-centric design. The series consists of two standards and a technical specification. For easy of reference the two standards and the technical specification will be referred to collectively as ISO/IEC 81346, but their full names are:

- ISO/IEC 81346-1:2009(E) Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 1: Basic rules (ISO and IEC, 2009a)
- ISO/IEC 81346-2:2009(E) Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 2: Classification of objects and codes for classes (ISO and IEC, 2009b)

- ISO/TS 81346-3:20012(E) Industrial systems, installations and equipment and industrial products – Structuring principles and reference designations – Part 3: Application rules for a reference designation system (ISO, 2012)

ISO/IEC 81346-1 and ISO/IEC 81346-2 are joint designation standards prepared by the IEC technical committee 3: Information structures, documentation and graphics symbols, in co-operation with ISO technical committee 10: Technical product documentation. ISO/TS 81346-3 was prepared by ISO technical committee 10: Technical product documentation, Subcommittee 10: Process plant documentation, and is based on *ISO/TS 16952-10*. The ISO/IEC 81346 series is the default option for forming reference designations in documentation for machinery sold within the EU as specified under EU Directive 2006/42/EC on machinery, unless otherwise agreed upon between manufacturer and user (Balslev, 2010). Many stakeholders within the engineering or other technical domains will therefore be familiar with the standard, and its concepts and principles. Using the standard as the basis for identification and reference in the models of an architecture description for production systems holds the following potential:

- The RDS system would be familiar to many stakeholders
- It will be easier to map between the RDS used in the design of the architecture to the documentation of the production system
- Industry IT-systems will be capable of attaching the codes to documents

The RDS presented here is both the result of analysis of ISO/IEC 81346 and sector specific reference designation systems based on the series; experiences from consultancy work with implementation of ISO/IEC 81346 in companies designing and building large processing plants; and my involvement in the development of a sector specific reference designation system for the Danish building industry. The RDS has also been tested in a procurement project in the primary case company to a lesser degree. It should be noted that the developed RDS not only provides a correspondence kind for use in interrelating models of the architecture description, it also constitutes a model in itself that describes some of the key structures of a production system or production system family. When taking this view of the RDS, it has proven useful to view the ISO/IEC 81346 standard in the context of an object oriented modeling paradigm to investigate the strengths and weaknesses of ISO/IEC 81346 for model correspondence in an architecture description.

29 The ISO/IEC 81346 standard series

ISO/IEC 81346 provides principles for structuring of objects including associated information and rules on forming reference designations based on the resulting structures. The standard is written so as to be applicable to almost any kind of technical system but is largely intended to be used in the context of large on-of-a-kind systems with many stakeholders. Because the scope of the standard is so large, it is not certain that it is directly applicable to the domain of architecture-centric design. The aim of the research is therefore to see if and how applicable the standard is as a means of model correspondence, and if it can serve as a basis for standardized data exchange in production system architecture descriptions, particularly in regards to design of multiple systems. Some issues have been discovered in the attempted application of the standard relating to the basic concepts of the standard, rules for forming reference designations and principles for structuring. An adaptation has been made of the standard that largely complies with the rules and principles set out, but with some modification to suit the domain of architecture description and design of multiple systems.

29.1 Core concepts of ISO/IEC 81346

ISO/IEC 81346 is based on three main concepts: Object, aspect and structure, which form the basis of the reference designations that can be defined and used for the purpose of identification and referencing. What follows is a brief introduction to the core elements of the standard for readers new to the standard.

Object

Objects are the elements of a system-of-interest referenced by a reference designation system, and the standard defines an object as an *"entity treated in a process of development, implementation, usage and disposal"* (ISO and IEC, 2009a, p.11). This definition is very broad and several examples are given throughout the standard to elaborate on the concept, e.g.:

> *"all kinds of objects and their constituents, such as plants, systems, assemblies, software programs, spaces, etc."* (ISO and IEC, 2009a, p.9)
>
> *"The object may refer to a physical or non-physical 'thing', i.e. anything that might exist, exists or did exist."* (ISO and IEC, 2009a, p.11)
>
> *"Elements of a system may be natural or man-made material objects, as well as modes of thinking and the results thereof (e.g. forms of organisation, mathematical methods, programming languages)"* (ISO and IEC, 2009a, p.12)
>
> *"the definition of the term 'object' is very general […] and covers all items that are subject to activities in the whole life cycle of a system"* (ISO and IEC, 2009a, p.13)
>
> *"there are objects that do not have a physical existence but exist for different purposes, for example:*
>
> - *An object exists only by means of the existence of its sub-objects, thus the considered object is defined for structuring purposes (i.e. a system);*
> - *For identification of a set of information"* (ISO and IEC, 2009a, p.13)

From the definition and all the examples provided it is clear that there is essentially no limit to what constitutes an object within the context of the standard, and to some extent it seems the standard attempts to allow for structuring and referencing of any information associated with the system-of-interest. In the context of this research project the system-of-interest is the production system, and the objects-of-interest are the design elements representing our view on the system i.e. the elements that are also modeled in the architecture description.

Aspect

An aspect is defined in the standard as a *"specified way of viewing an object"* (ISO and IEC, 2009a, p.12), and is regarded as a sort of filter to *highlight* the information that is of relevance for an object. The aspects are used for the purpose of defining object structures. Three basic aspects are defined in the standard:

- *"what an object is intended to do or what it actually does – the function aspect;*
- *By which means an object does what it is intended to do – the product aspect;*
- *Intended or actual space of the object – the location aspect"* (ISO and IEC, 2009a, p.14)

Each aspect is associated with a prefix used in the forming of reference designations (see Table 7). It is possible to use other aspects not defined in the standard for the purpose of structuring if the three basic aspects are not applicable or sufficient to describing the system-of-interest.

Table 7 - Prefixes for aspects

Prefix	Aspect
= (equals)	Function
- (minus)	Product
+ (plus)	Location
# (number)	(Other aspects)

Structure

Structures are used in the standard to organize objects in relation to each other, and structure is defined as the *"organization of relations among objects of a system describing constituency relations (consists-of / is-a-part-of)"* (ISO and IEC, 2009a, p.12). The structures describe hierarchical relations between objects and an element in the structure may be defined as being a constituent of only one other element within the same structure (see Figure 70). If a reference designation system is seen as a form of object oriented model of a system-of-interest then the constituency concept of ISO/IEC 81346-1 therefore describes composition and not aggregation relationships between the modeled objects.

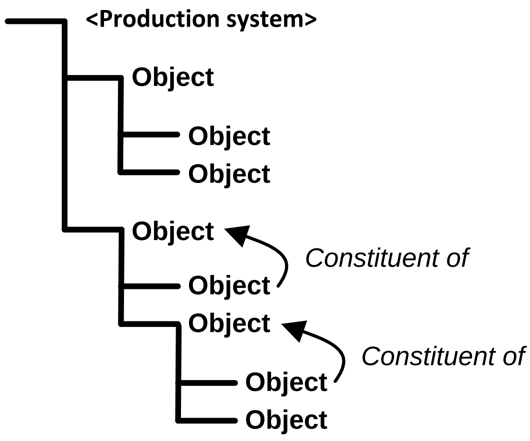


Figure 70 - Structuring of objects in ISO/IEC 81346

Structures are formed based on the aspects and more than one aspect may be applied in a structure. This means that it is possible to change the applied aspect between levels of the structure, so that a level is formed based on another aspect than the preceding higher level in the structure. Structures that are only formed based on a single aspect are referred to as aspect-oriented.

The elements that are identified in the structures are referred to as occurrences, which is the same as instances of an object class in an object oriented paradigm. The standard refers to individuals as the actual real world object that is modeled in the RDS, e.g. a physical unit with a serial number. The individual exist in the real world, where as the instance is a model object.

Reference designations

Reference designations are seen as descriptions of an objects address/place within the structures that serve as references for the objects. The reference designation that identifies the object is formed by first assigning the object a single-level reference designation that is unique with respect to the object of which it is a constituent, and then forming a multi-level reference designation by concatenating the single-level reference designations down through the structure from the top-most object down to the object-of-interest (see Figure 71).

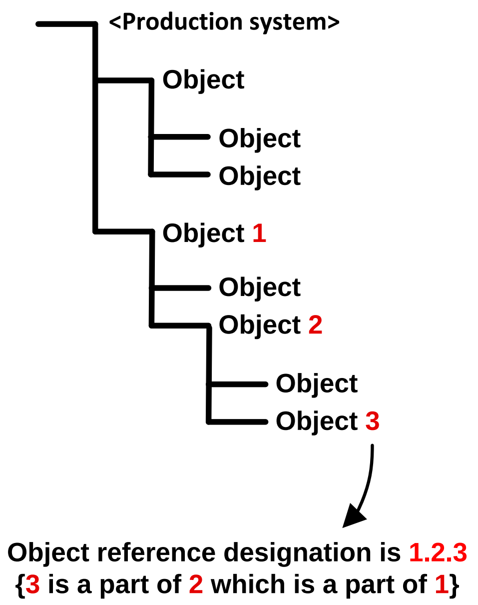


Figure 71 - Forming reference designation by concatenation.

Single level reference designations consist of a prefix representing the aspect under which the object is considered, followed by either:

- a letter code representing a classification of the object
- a number to distinguish between objects
- a letter code followed by a number

The reference designation is constructed as follows:

A Classification of the structured object is represented by a letter code consisting of one or more capital Latin letters depending on the depth of the classification. In letter codes with more than one letter the second (third, etc.) letter indicates a sub-class of the class represented by the preceding letter (see Figure 72). The letters "I" and "O" are mitted because they can be confused with the numbers one and zero.

A1 Numbers are used to distinguish between objects of the same class. Classification may also be omitted and only numbers used to identify the object.

=A1 A prefix is used to indicate the aspect applied in the formulation of the reference designation. The sign is chosen from the G0-set of ISO/IEC 646.

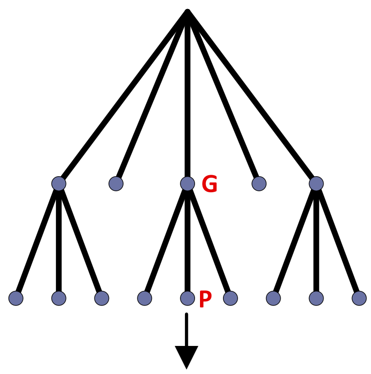


Figure 72 - Classification follows a compositional hierarchy where letters for the subclass are added to the superclass.

A multi-level reference designation serving as the identifier of an object is constructed as follows:

=A1=P1 A multi-level reference designation is formed by concatenating the single level reference designations down through the structure to the structured object. The multi-level reference designation thus both serves as an identifier and a representation of the structural context of the object.

=A1.P1 **or** **=A1P1** If the prefix sign for two single- level reference designations is the same, then the prefix sign between them may be replaced by a full stop, or completely omitted if the preceding reference designation ends with a number and the following reference designation begins with a letter.

A reference designation is said to be unambiguous if it describes the exact address of the object-of-interest in the structure or ambiguous if it does not directly reference the object-of-interest's address in the structure but rather a higher level object of which the object-of-interest is a part.

Multiple reference designations

The standard specifies that an object may appear in different structures e.g. in a product oriented structure and a location oriented structure. In this case the object may be referenced both by its reference designations from each structure or by forming a reference designation set consisting of the individual reference designations ascribed to the object. Each reference designation of the set will identify the object, and no other object.

=A1/–M1 A reference designation set can be written in a single line or with each of each of the reference designations on separate lines. When writing the reference designations in a set on the same line they are separated by a / (solidus).

It is possible to define more than one structure based on an aspect, in order to represent different structures of the system for different purposes. Such alternative structures are represented by using multiple prefixes of the aspect in the reference designations, e.g. ++, +++, +++ and so forth.

Specific designations

In addition to referencing the constituent elements of the system-of-interest part three of the standard (ISO/IEC 81346-3) allows for the specific designation of signals (interaction between objects in the form of information exchange), terminals (interfaces) and documents associated with the objects. All are identified with respect to the object to which they belong or are assigned, and are given a prefix (see Table 8). The specific designations are formed following the rules of ISO/IEC 81346-3.

Table 8 - Prefixes for specific designation

Prefix	Task
; (semi colon)	Signal
: (colon)	Terminal
& (ampersand)	Document

The specific designations are formed by adding a reference designation for the signal, terminal or document to the end of the object reference designation as seen below:

=A1;S1 Specific designation for a signal assigned to object =A1.

–K1:U1 Specific designation for a terminal assigned to object –K1.

–K1&E1 Specific designation for a document assigned to object –K1.

A reference designation is not assigned to the topmost node in a structure i.e. the node/object representing the entire system-of-interest. An identifier can however be assigned if the top node is to written together with a reference designation. The identifier is written between angle brackets “<>” e.g. <A1>. A reference designation could then be written as <A1> =G1.

29.2 Limitations in ISO/IEC 81346

From a review of the standard series several limitations or flaws in the concepts and rules have been identified. What follows is a short explanation of some of the key issues.

Object definition

From reading the ISO/IEC 81346 standard series it is clear that it does not employ a consistent perception of the object concept. At times the standard is very specific in regards to what the objects are, and at other times objects are defined so broadly that they could be anything, indeed any entity that could possibly be seen as a part of the system as long as it is treated in a process of development, implementation, usage and disposal. In contrast the standard often treats objects as physical entities that exhibit functionality and have a location, and yet the object definition is so broad that this is not the full extent of the objects that can be referenced. This vagueness and uncertainty as to what objects the standard can handle certainly does not aid users of the standard, particularly because there is no requirement in the standard for defining the structured objects when the standard is applied in a specific context.

Uncertainty of structured objects

Aspects are applied in structuring of sub-objects of an object-of-interest, but it is not fully explained what an aspect signifies or how the sub-objects are to be defined. It is said *that “only constituent objects (i.e. sub-objects) are seen that are relevant in that aspect”* (ISO and IEC, 2009a, p.14). For the function aspect this means that only sub-objects that are relevant to what the object-of-interest (the deconstructed object) *“is intended to do or what it actually does”* (ISO and IEC, 2009a, p.14) are included in structures based on a particular aspect. This explanation is open to interpretation. For a structure formed in regards to the function aspect the sub-objects could for example be objects that define functionality (as the standard defines function), exhibit functionality, facilitate functionality, etc. An example would be that objects in a function-oriented structure could be either processes, physical parts carrying out a process or locations where processing takes place, as long as it is relevant to reference them based on a processing perspective of the system-of-interest. This uncertainty is further emphasized by the fact that the standard suggests that the same object can be structured under different aspects and that separate objects structured under different aspects can potentially be merged at some point in the life-cycle if they can be considered to be the same object. It stands to reason that it is not directly obvious to users of the standard what kinds of objects are referenced by reference designations formed from the structures, unless the implementation of the standard specifies what object types are structured under each aspect. The need to define structured objects is however not a requirement that the standard addresses, but examples of this need for definition is demonstrated by ISO/TS 16952-10, which does to some extent define the structured objects for power plants. The conclusion is that the aspects themselves do not define the kinds of objects structured and that unless the users provide a definition it is very uncertain what exactly is being referenced.

Aspect transitions

There is limited applicability in the transition from one aspect to another in a structure, since this requires a 1:1 relation to exist, and that the containing object (structured according to one aspect) is compatible with the contained object (structured according to another aspect) i.e. can a process for example contain a location?

Describing generic structures

One of the key limitations of the standard is that it is intended to be applied in the structuring of individual systems-of-interest, not groups of systems, and that it is intended to be used to reference specific system compositions and not generic design structures, that provide the basis for multiple systems. This is seen by what is structured and the assigned reference designations. The standard structures object occurrences (i.e. object instances), and not classes of objects. The structures and reference designations therefore do not include any direct way of addressing cardinality but seeks instead to identify each instance of an object class within a system. This presents a challenge in describing generic structures where objects may be instantiated multiple times to form a specific system design, e.g. in the case where an injection molding machine is replicated in a production line to increase production capacity. The standard as it is described would include each instance of the machine in the structure and assign them separate reference designations. But the architecture description includes a description of the generic structure, where the production line simply consists of a machine design (an object class) that has cardinality. If the standard is used for model correspondence in an architecture description there is therefore a need to reference not just each instance of the machine used in the production system, but also the machine design that has been used multiple times in the system.

Expressing variety of system elements

Regardless if the referenced elements are processes, organs, parts, locations or any other kind of object, there is a need within an architecture description to be able to identify variants of the objects. The only means for describing variety in the standard is by either forming different structures, including objects representing each variant or through a more detailed classification scheme that would include classes for each object variant. None of these options are particularly desirable, either because they require structuring of too many objects and formulation of different structural configurations, or because they strain the classification scheme. The specified classification tables of ISO/IEC 81346-2 only include two levels in the classification scheme, but to represent variants of the classified objects in a design, it could very well be necessary to use many more e.g. four or five letters. The classification scheme would also very easily grow quite large as new variants of system elements, e.g. new parts are added to the architecture.

Ambiguous reference designations

The possibility of using an ambiguous reference designation (e.g. -M1.K1…) as part of a reference designation set (=P1.GP1/-M1.K1…) means that the ambiguous reference designation does not constitute a direct identifier for the object to which the reference designation set belongs. Rather the ambiguous reference designation is an expression of a relation between the referenced object (=P1.GP1) and a higher level object in another aspect (-M1.K1). Reference designation sets that include ambiguous reference designations therefore not only identify objects, they also include references for the objects relations to other objects.

Object traceability throughout life-cycle

The standard claims to allow *"old structures to be handled together with new structures by using multiple unambiguous identifiers"* (ISO and IEC, 2009a, p.8), but no guidelines or rules for this are provided. If the intention is merely to assign multiple reference designations to the same object as part of a reference designation set, then I would claim that it would still be impossible to see which structure a particular reference designation belonged to. This would not provide traceability between the old and new structures, unless either:

- Each structure was given a new prefix, e.g. + (old structure), ++ (new structure)
- Each structure was assigned a new top node identifier, e.g. <A1> (old structure), <A2> new structure>
- No objects were ever removed from the structure.

All three of these options would eventually inflate the number of reference designations to an unreasonable degree. In general it is observed that the standard in its current form does not provide for tracking of changes in system structures, and this must be assumed to be handled by separate documentation or IT-systems which can provide mapping between different versions of the structures in the RDS.

30 Sector specific reference designation systems

Sector specific reference designation systems based on the ISO/IEC 81346 standard series have also been studied as part of the review of the standard. The sector specific reference designation systems that have served as input are:

- ISO/TS 16952-10 (power plant sector)
- Cuneco Classification System (construction sector)

What follows is a brief summation of lessons that have been learned from studying of ISO/TS 16952-10 and from my participation in the development of the identification elements of the Cuneco Classification System (to be released in 2014).

30.1 ISO/TS 16952-10 (RDS-PP)

ISO/TS 16952-10: The technical *specification ISO/TS 16952-10:2008(E) Technical product documentation – Reference designation system – Part 10: Power plants* (referred to as *RDS-PP*) is a technical specification that describes an RDS for use in systems for industrial production of electrical and thermal energy i.e. power plants or elements of power plants. The system can be used for referencing both technical objects and their associated documentation. A sector-neutral version of the specification (ISO/TS 81346-3) is currently under consideration for incorporation in the ISO/IEC 81346 standard series as a basis for ensuring a consistent interpretation of the RDS rules in ISO/IEC 81346-1.

The RDS-PP is of interest because of the resemblance between a power plant and a production system for production of products. Both types of systems are a kind of processing plants. A review of the RDS-PP shows how the ISO/IEC 81346 could be applied for systems resembling production systems and it serves as inspiration in the formulation of the PSAF-RDS. The following summarizes some of the key learnings from the review:

- The object definition is more restricted in RDS-PP. There is an attempt to specify more precisely, what kinds of objects are structured under each aspect.
- There is no use of reference designation sets, i.e. objects are not present in multiple structures.
- Only transition from the function aspect to the product aspect is permitted. Transitions signify a 1:1 relation between objects.
- There is a predefined maximum of structural levels with classification tables associated to each structural level.
- The standard uses the designations seen in Table 9.

Table 9 - Designations used in RDS-PP with translation to the concepts used in PSAF.

Prefix	Designation task/aspect in RDS-PP	Designation task/aspect in PSAF context
#	Conjoint designation	System-of-interest
=	Function-oriented designation	Organs/Functional elements
==	Functional allocation	Technical processes
+	Point of installation	Spaces defined by parts
++	Location	Spaces
-	Product-oriented designation	Parts
:	Terminal designation	Interfaces
;	Signal designation	Signals
&	Document designation	Documents

In general it can be said that the RDS-PP technical specification rectifies some of the problems with the ISO/IEC 81346 standard. This is especially in regards to the concept definitions used throughout the standard (even though there is still some flexibility in the concepts) and in the fact that objects are not structured in multiple structures. The separation between the function-oriented designation and the functional allocation also demonstrates a realization of the difference between the concept of processes and functional elements.

30.2 Cuneco Classification System (CCS)

The Cuneco Classification System (CCS) is a system for classification of information in the Danish building sector and it includes among other a reference designation system for structuring and referencing building elements and spaces. CCS is the replacement for the Danish building classification 2006 (DK: Dansk Byggeklassifikation 2006) expected to be launched sometime in 2014. Both DBK2006 and CCS are developed by the industry association *bips*, which is responsible for developing standards, working methods, tools and sector and industry standards for the construction sector as part of a Danish government imitative for increasing the use of information and communications technology in the Danish construction sector, known as *Digital Construction* (DK: Det digital byggeri). I have been a part of the core working group responsible for developing the coding and structuring principles and rules for the included RDS for referencing elements and spaces of construction complexes. This work has been based both on the ISO/IEC 81346 standard series and additional classification systems. The coding and structuring principles have been the subject of two public hearings where both public and private institutions, companies and individuals have provided input for the responses and input. The working group's reports as well as comments from the public hearings and the responses to them can be found at *http://cuneco.dk/ccs-kodestruktur*.

While the CCS is intended to be used in the Danish construction sector it bears comparison with the task of correspondence in a production system architecture description. The CCS is intended to be used by both suppliers, designers, builders and users of construction complexes throughout the entire life-cycle, and as such it must address the needs of a multitude of stakeholders. Building works themselves can also be thought of as large technical systems that include mechanical, electrical and structural elements, and it is reasonable to compare them to production systems in many respects. The following summarizes some of the key learnings from my participation in the working group dealing with the coding and structuring principles for building elements and spaces and is primarily based on the report presented at the second public hearing (cuneco, 2012) and the third release version of the coding and identification principles of CCS, *CCS Identifikation (R3)* (cuneco, 2014).

Dynamic structures

The structures described by the RDS are dynamic in several regards:

- When the structures are initially formed, they may be formed as both a top-down structuring where the design is progressively made more detailed, or they may be formed through bottom-up structuring in which objects are collected into wholes.
- The structures will change through-out the life-cycle of the system-of-interest, both as objects are added and deleted, and as objects may see changes in their relations.

The dynamic nature of the structures can make it difficult to maintain traceability of an object through-out the life-cycle without the use of a system for mapping between different versions of the structures. The ISO/IEC 81346 does not specify any rules or principles for maintaining traceability between structures at different time in the life-cycle. CCS seeks to address this partially by way of the RDS itself, and by suggesting that stakeholders employ IT-systems with included versioning for maintaining the structures. The way that CCS itself can maintain some form of traceability is by employing a global numbering for each object class (% aspect). This way, even if the object's relations change and it is assigned a new place in the multi-level structure (e.g. ==M2.GP37) the global numbering will allow for mapping between the old and new structure. The numbering may also be repeated in different structures based on the same aspect, e.g. =GP*37* (single level structure), and ==M1.GP*37* (multi-level structure).

Spaces

The parts of a technical system are mostly perceived as material objects with a primarily solid state, but it can be equally important to think of parts with other primary matter states e.g. liquid, gas or plasma states. In CCS this is exemplified by the desire to not only identify the solid material elements of a building but also the so-called spaces. Spaces in CCS are not merely locations, they are in themselves an element of a building (and one of the most important at that), and they are as much a design element as the bricks, girders, windows, valves, generators, etc. that make up a construction complex. In the context of design of production systems it should be considered if these kinds of elements are adequately covered by the Part concept, or if they need to be addressed separately? As an example, for some design purposes it would not only be relevant to identify the equipment within a production cell, but it would also be relevant to identify the cell space for the purpose of specifying requirements for access, safety, noise levels, humidity and temperature conditions, etc. It could well be that a stakeholder holds a view of the production system as a collection of spaces in which processes take place, e.g. storage spaces, processing spaces, transport spaces and so on, and that these spaces are treated as elements in the activities of design.

Classes and instances

The ISO/IEC 81346 identifies objects (*instances*) and their *composition* relations. The *instance* relation to the object class is shown by the use of a classification code (letter code). CCS has demonstrated a need to identify object *classes* and their *generalization/specialization* relations in addition to the instances, i.e. CCS also specifies the need to identify design variants. This is to be used for quantity analysis in the design. Determining the quantities of variants used in a particular construction complex will among other serve to support procurement and maintenance of the completed complex.

Changing classification

If the classification of an object is refined throughout the life-cycle of the object this will change the reference designation of the object, which is equivalent to defining a new object. This means that for traceability's sake, the classification of the object is locked when the object is first defined. Depending on which classification scheme is applied this can be problematic in a process of design, as designers may initially not be able to assign a detailed classification to the object which might be necessary later in the object life-cycle.

31 RDS concepts

This section describes the basic characteristics of the RDS included in PSAF. The requirements are based on the identified needs of model correspondence in the viewpoints of PSAF and the reviews of the ISO/IEC 81346 and sector specific standards. The requirements cover:

- Identified architecture elements
- Aspect definitions
- Structuring possibilities

Identified architecture elements

The RDS will be used to reference the key elements of the production system that are modeled in the different model kinds of PSAF. These include:

- Operators
- Technical process

At the moment the only operator that is described in PSAF is the Technical system so for now the PSAF-RDS therefore only covers the Technical system and will allow for referencing of the following elements of the production systems:

- Technical processes and their constituent structures
- Organs and their constituent structures
- Parts and their constituent structures

Aspect definition

Modeling merely by aspects makes it unclear what kind of object is referenced, and the resulting reference designation itself does not describe the kind of object i.e., the fundamental nature of the object. This is fundamentally undesirable if elements of the architecture are to be identified explicitly and unambiguously, i.e. if users are to know if a reference designation identifies a process, organ or part. In the RDS the aspects therefore define specific object domains. This means that objects structured in regards to an aspect are constituents of the domain and the reference designations constitute identifiers of the objects of the domain i.e. organs, parts, activities/processes. This also means that the RDS does not reference so-called merged objects and that an object only exists in structures formed under one aspect. The aspect definitions are as follows:

Process aspect: The process aspect represents the activity/process domain defined in theory of technical systems. It is used to structure and identify the technical processes of the production system. The aspect is roughly equal to the function aspect from ISO/IEC 81346 (when it is applied in a process focused way) and the functional allocation designation from ISO/TS 16952-10. It is chosen to use the prefix for alternative aspects (#, number) as allowed by ISO/IEC 81346 in order to distinguish between the three domains.

Organ aspect: The organ aspect represents the organ domain from theory of technical systems. The aspect is used to structure and identify the organs of the technical system of the production system. The aspect is roughly equal to the function aspect from ISO/IEC 81346 when the standard applies the aspect in a system oriented manner. The aspect could also be referred to as a function aspect or system aspect since it describes the functional sub-systems of the production system.

Part aspect: The part aspect represents the part domain from theory of technical systems. It is used to structure and identify parts in the technical system of the production system. The aspect is roughly equal to the product aspect from ISO/IEC 81346, when the standard applies the aspect in a component oriented manner.

The prefixes of the aspects can be seen in Table 10 - Prefixes and aspects of the PSAF RDS.:

Table 10 - Prefixes and aspects of the PSAF RDS.

Prefix	Aspect	Domain
#	Process aspect	Activity domain
=	Organ aspect	Organ domain
-	Part aspect	Part domain

While it is possible to form multiple different structures under each aspect, as there are many different ways of expressing the constituent relations within a domain, stakeholders are advised to define one primary structure under each aspect to provide the main reference for the elements across all models of the architecture description. Alternative structures can still be used to capture and communicate the many different structures expressed in the architecture description.

Structures of architecture elements

The referenced elements of the domains are related to each other in constituent structures, and the model kinds of the architecture description is capable of describing both the generic structures of elements (e.g. the generic organ structure or generic part structure) and the specific structures with all instances of the design elements (e.g. a specific system configuration with all instances of a specific part) (see Figure 73).

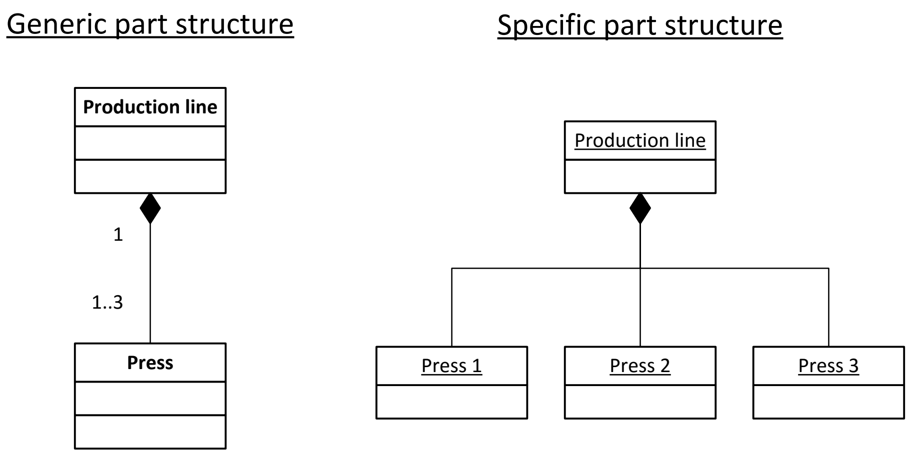


Figure 73 - The system structure can be expressed both through a generic structure showing the principle of the design and a specific structure showing the specific makeup of the system.

The ISO/IEC 81346 is not intended to be directly able to reference elements in a generic structure, which is then later instantiated in the system configurations of different production systems. The reason for this is that the standard does not operate with a class concept in the way that object oriented modeling does. The standard structures what it calls *Occurrences* and assigns classification to them. In the terminology of object oriented modeling the classes used in classifying the occurrences are in actuality meta-classes and the structured elements are class instances (see Table 11).

Table 11 - Relation between concepts in ISO/IEC 81346 and an object oriented modeling paradigm.

ISO/IEC 81346	Object oriented modeling
Class	Metaclass
	Class
Occurrence	Instance (object)
Individual	

If ISO/IEC 81346 was applied directly without addition of the class concept then the structures described by the RDS would contain all possible instantiations of the production system design with codes for all the constituent elements. This is problematic because:

- It provides an excessive amount of reference designations for systems with many different configurations. In short the structures would have to reflect all possible configurations.
- It does not support referencing elements representing multiple elements in a specific system e.g. if a process can be multiplied in the system e.g. for parallel processing on different equipment, then each instance of the process would have a reference designation identifying the specific instance. There would be no reference designation identifying the basic process which had been duplicated.

In order for the RDS to be able to describe generic structures the class concept is therefore added. This addition also allows for referencing of variants of the elements, e.g. referencing of part variants.

Two structuring principles

Referencing of either generic or specific structures is realized by applying two separate structuring principles in the RDS. The two structuring principles will provide distinct sets of reference designations that can be used as references for either the generic design or specific design. Stakeholders may choose to apply either structuring principle or only one depending on the needs for model correspondence in the architecture description. The two principles are defined as follows.

Design RDS: Design RDS provides reference designations based on the generic design structures of the production system or production system family. The structures expressed through the reference designations are the generic structures of the production system's technical system that describe the principle element of the technical processes system, organ system and part system incl. variants of the elements of the systems, e.g. process variants, part variants, etc. The elements of the structures are the object classes representing the design elements of the generic structures. The structures are typically expressed in the models of the architecture description and do not have to be defined separately. For example the generic process structure can be found in the PCV-2, and the generic organ structure and generic part structure can be found in the engineering view and part view of the TSV-4 respectively.

System RDS: System RDS provides reference designations based on the specific design structure of the production system or production system family. The structures expressed through the reference designations are the specific structures of the production system's technical system that include all instances of processes, organs and parts included in the instantiated design. The System RDS can be used to reference all instances of processes, organs and parts in a specific production system based on the architecture.

If both Design RDS and System RDS are used to provide reference designations in the architecture description, because both the generic design structures and specific design structures need to be referenced, then an identifier for the top node of the structures is included as a prefix for the reference designations:

- Identifier for Design RDS: <D>
- Identifier for System RDS: <S>

The main characteristics of the two structuring principles are summarized in Table 12 below:

Table 12 - Main characteristics of the two alternatives for structuring in the reference designation system.

	Design RDS	System RDS
Top node	<A>	<S>
Structured elements	Object classes	Object instances
Structure type	Generic design structures	Specific design structures
Aspects	# Technical process = Organ - Part	
Classification	ISO/IEC 81346-2:2009(E)	
Specific designations	ISO/TS 81346-3:2012(E)	

To better understand the need for referencing of generic and specific structures consider the following example of a process flow where an assembly process can be duplicated, e.g. to obtain a higher production capacity or redundancy. In this case the architecture description might describe both the generic process flow and the specific process flow representing the full possible configuration. In this case there could be a need to reference both the generic process and the two specific instances of the process within the architecture description, it could look like this:

Structuring according to Design RDS

Figure 74 shows a generic process flow for a production system. Structuring according to Design RDS provides reference designations for the classes of processes but not all possible instances of the processes. A sub-process in the process flow has two variants that allow for different processing of operands. Both the sub-process and its two variants can be referenced separately.

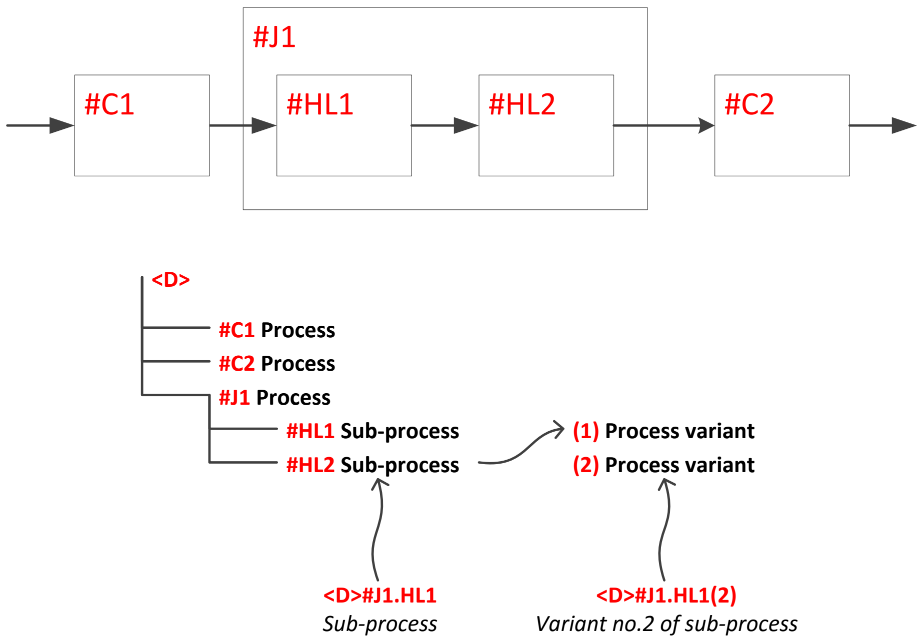


Figure 74 - Example of Design RDS coding in the process aspect.

Structuring according to System RDS

Figure 75 shows the maximum possible configuration of processes based on the generic process flow. Some processes of the generic flow can be duplicated in the production system in order to provide extra capacity and redundancy. In the example the process #J1 has been duplicated and both instances of the process are structured and assigned reference designations.

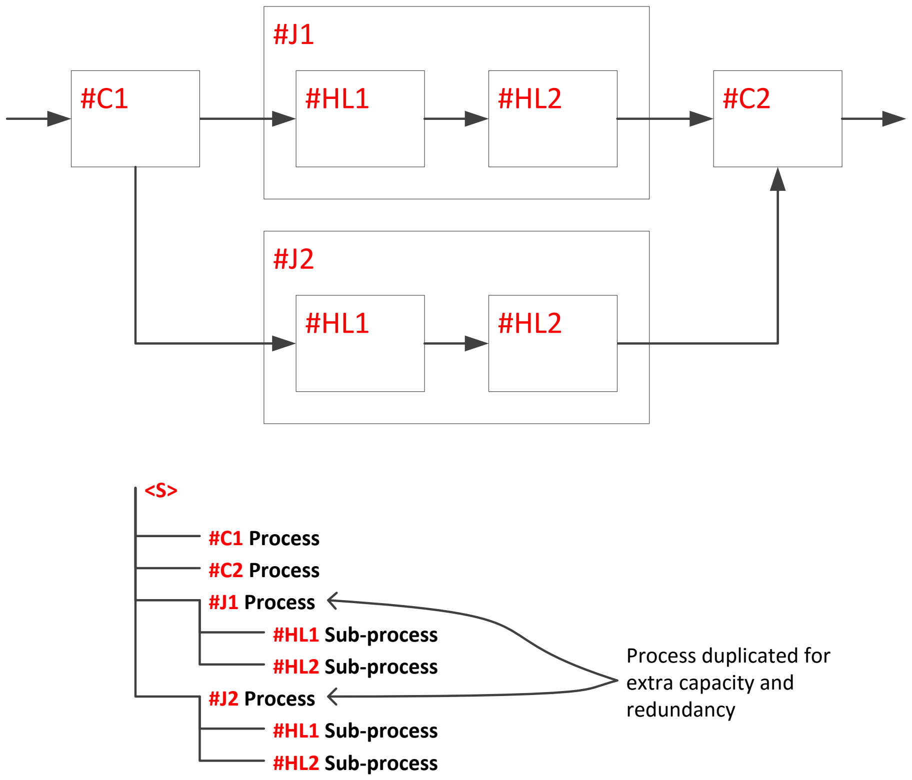


Figure 75 - Example of System RDS coding in the process aspect.

The reference designations of both Design RDS and System RDS can be freely used in the different models of an architecture description where ever there is a need to reference either a process, organ or part.

32 RDS coding syntax

The RDS follows the same rules for forming single and multi-level reference designations as defined in ISO/IEC 81346-1, and the classification tables and classification principles from ISO/IEC 81346-2 are also applied to provide meta-classes of the structured objects.

An addition to the numbering syntax is made in order to identify variants of elements under the Design RDS structuring principle. Structures formed with respect to the Design RDS structuring principle contains object classes representing the design elements of the system, and variants of the elements constitute instances of the structured object classes. (García and Gelle, 2006) suggests using a number in parenthesis at the end of the reference designation to identify instances of the referenced object. This practice is adopted into the RDS to instead identify variants of the structured objects. Referring to the class is then done by the reference designation without cardinality, and referring to the instances of the class is done through the addition of a number in parenthesis for the instance. An example of a pump in the production system and its two variants can be seen below:

-GP1	Pump 1
-GP1(1)	Variant 1 of Pump 1
-GP1(2)	Variant 2 of Pump 1

What follows is the specific coding syntax for reference designations in the three aspects.

Process aspect

The structures described in the process aspect can either be restricted to the main technical processes being performed by the technical system, in which case the generic process structure can be found in the PCV-2 process flow diagram, or they may include all processes of the technical system including the supporting processes. The classification of the processes follows table 2 or 3 of the ISO/IEC 81346-2 depending on the need of the stakeholders for classification. Table 3 is intended to be adapted by the users to the specific production domain where the architecture description is applied. Table 2 defines classes that are independent of the application domain i.e. the classes can be applied in different kinds of production. The following defines the coding syntax for forming reference designations under the process aspect.

Design RDS syntax

Breakdown level			1		n			
Number / type of data position	**<D>**	**#**	**C**	**N**	**.**	**C**	**N**	**(N)**
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)

B **Classification letters**

N **Numbering**

(0) ID of structuring principle
(1) Aspect prefix
(2) Classification according to ISO/IEC 81346-2 Table 3
(3) Subdivision of part e.g. module, assembly, component, etc.
(4) Breakdown mark, either aspect prefix, "." (period) or no character.
(5) Classification according to ISO/IEC 81346-2 Table 2
(6) Subdivision of part e.g. module, assembly, component, etc.
(7) Variant numbering. Only included when identifying process variants.

System RDS syntax

Breakdown level			1		n		
Number / type of data position	**<S>**	**#**	**C**	**N**	**.**	**C**	**N**
	(0)	(1)	(2)	(3)	(4)	(5)	(6)

B **Classification letters**

N **Numbering**

(0) ID of structuring principle
(1) Aspect prefix
(2) Classification according to ISO/IEC 81346-2 Table 3 or 2
(3) Subdivision of technical process.
(4) Breakdown mark, either aspect prefix, "." (period) or no character.
(5) Classification according to ISO/IEC 81346-2 Table 2
(6) Subdivision of technical process

An example of reference designations applied in the PCV-2 can be seen in Figure 76. The figure shows a generic process flow for an assembly system. The reference designations are formed based on Design RDS.

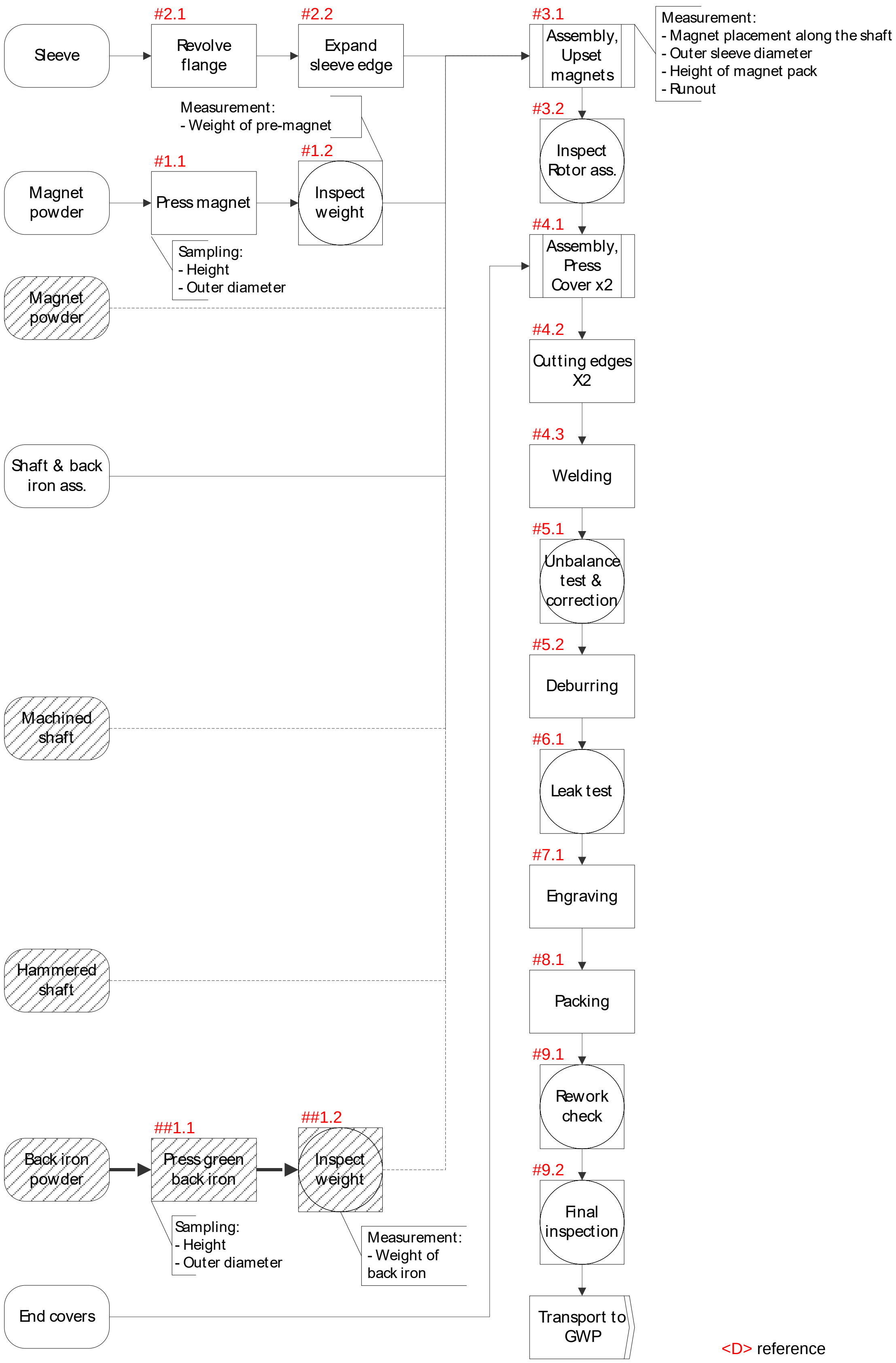


Figure 76 - Reference designations applied to the generic process flow in PCV-2 for an assembly system.

Organ aspect

The organ structures describe the functional sub-systems of the technical system and can either be found in the Engineering view of the TSV-4 Technical System Master Plan or in the TSV-2 Interface diagram. The classification of the organs follows Table 3 of the ISO/IEC 81346-2 for the first structuring level and table 1 or 2 for subsequent breakdown levels. The following defines the coding syntax for forming reference designations under the organ aspect.

Design RDS syntax

Breakdown level			1		n			
Number / type of data position	**<D>**	**#**	**C**	**N**	**.**	**C**	**N**	**(N)**
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)

B **Classification letters**

N **Numbering**

(0) ID of structuring principle
(1) Aspect prefix
(2) Classification according to ISO/IEC 81346-2 Table 3
(3) Subdivision of organ
(4) Breakdown mark, either aspect prefix, "." (period) or no character.
(5) Classification according to ISO/IEC 81346-2 Table 1 or Table 2
(6) Subdivision of organ.
(7) Variant numbering. Only included when identifying organ variants.

System RDS syntax

Breakdown level			1		n		
Number / type of data position	**<S>**	**#**	**C**	**N**	**.**	**C**	**N**
	(0)	(1)	(2)	(3)	(4)	(5)	(6)

B **Classification letters**

N **Numbering**

(0) ID of structuring principle
(1) Aspect prefix
(2) Classification according to ISO/IEC 81346-2 Table 3
(3) Subdivision of organ.
(4) Breakdown mark, either aspect prefix, "." (period) or no character.
(5) Classification according to ISO/IEC 81346-2 Table 1 or Table 2
(6) Subdivision of organ.

An example of reference designations applied in the TSV-4 can be seen in Figure 77. The figure shows a section of the engineering view in a TSV-4 for a tester system. The reference designations are formed based on Design RDS and include references for organ variants.

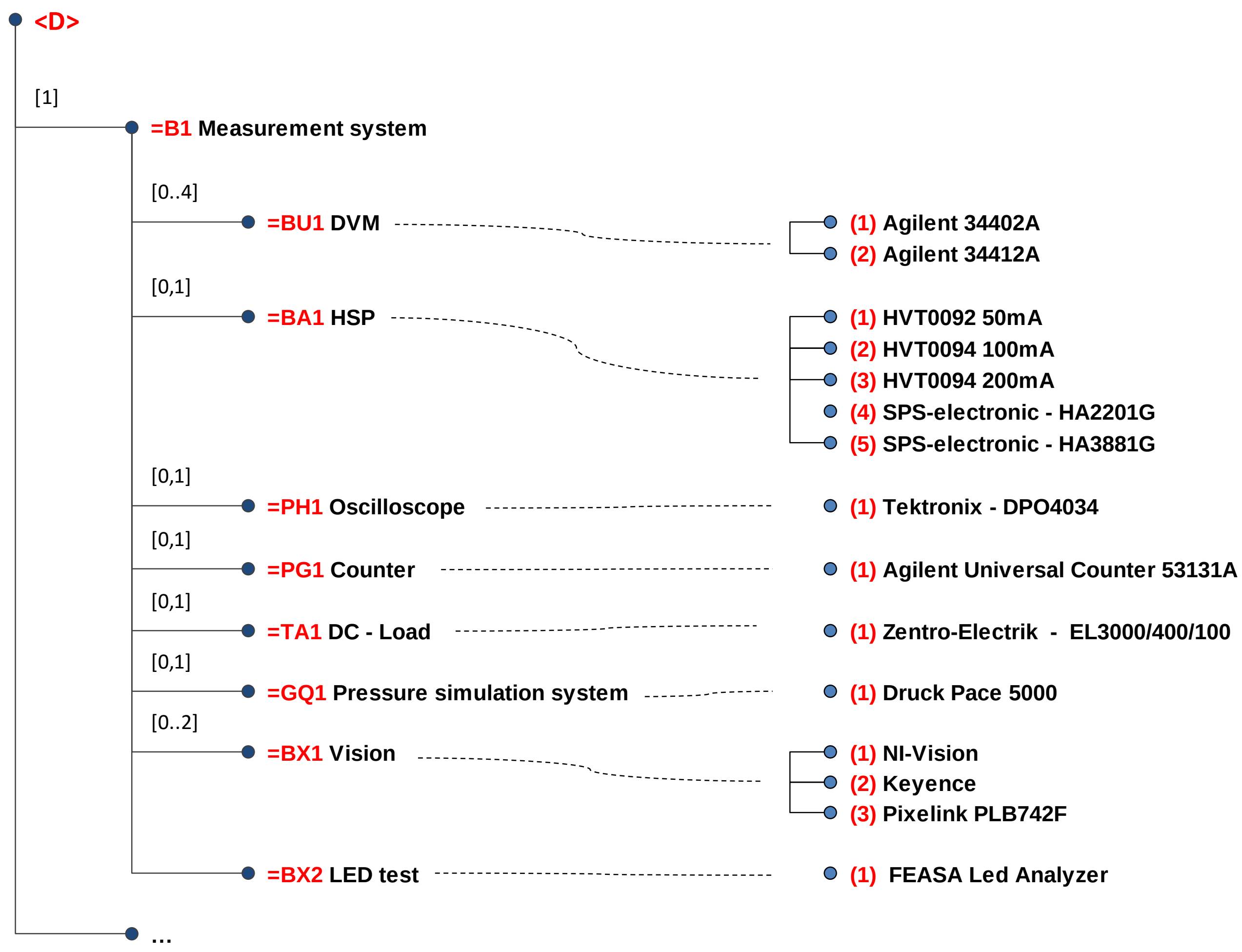


Figure 77 - Reference designations applied to the engineering view of TSV-4 for a sub-section of a testing system.

Part aspect

The part structures describe the functional part design of the technical system and can either be found in the Part view of the TSV-4 Technical System Master Plan or in the TSV-2 Interface diagram. The classification of the parts follows Table 1 and table 2 of the ISO/IEC 81346-2. The following defines the coding syntax for forming reference designations under the organ aspect.

Design RDS syntax

Breakdown level			1		n			
Number / type of data position	**<D>**	**#**	**C**	**N**	**.**	**C**	**N**	**(N)**
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)

B Classification letters

N Numbering

(0) ID of structuring principle
(1) Aspect prefix
(2) Classification according to ISO/IEC 81346-2 Table 1 or Table 2
(3) Subdivision of part e.g. module, assembly, component, etc.
(4) Breakdown mark, either aspect prefix, "." (period) or no character.
(5) Classification according to ISO/IEC 81346-2 Table 1 or Table 2
(6) Subdivision of part e.g. module, assembly, component, etc.
(7) Variant numbering. Only included when identifying organ variants.

System RDS syntax

Breakdown level			1		n		
Number / type of data position	**<S>**	**#**	**C**	**N**	**.**	**C**	**N**
	(0)	(1)	(2)	(3)	(4)	(5)	(6)

B Classification letters

N Numbering

(7) ID of structuring principle
(8) Aspect prefix
(9) Classification according to ISO/IEC 81346-2 Table 3
(10) Subdivision of part e.g. module, assembly, component, etc.
(11) Breakdown mark, either aspect prefix, "." (period) or no character.
(12) Classification according to ISO/IEC 81346-2 Table 1 or Table 2
(13) Subdivision of part e.g. module, assembly, component, etc.

An example of reference designations applied in the TSV-2 can be seen in Figure 78. The figure shows a configuration layout for an injection molding cell and two referenced equipment variants. The reference designations are formed based on Design RDS. The same reference designations can be found in among other the TSV-4 thereby creating a link between the models.

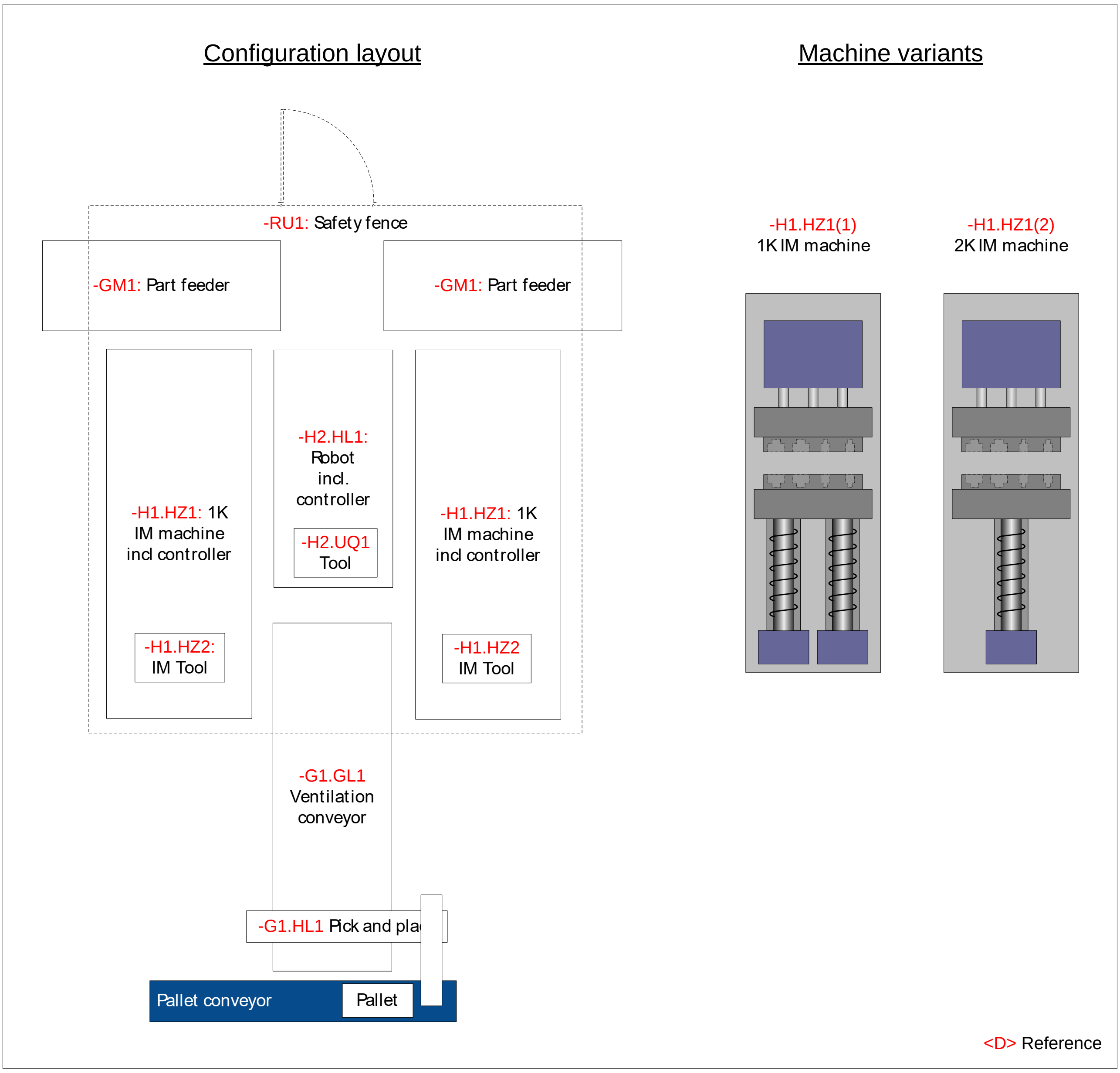


Figure 78 - Reference designations applied to the configuration diagram of an injection molding cell. Two equipment variants are shown next to the configuration layout.

33 Conclusion on RDS for architecture description

This chapter introduced a reference designation system that can be used to provide referencing of objects that are referenced in multiple models of the architecture description, or in documents, systems or processes not related to the architecture description. The suggested RDS is the result of a study into the applicability of the ISO/IEC 81346 standard series as a support for information handling in architecture-centric design. It was found that the standard has some limitations in its expressive capability and that there are some fundamental problems with the applications of object concepts in the standard. However it was also found that the standard can be adapted to referencing of key structures and elements of the production system, namely technical processes, organs and parts, and their constituent structures within the system. With some modification to the basic concepts of the model, including the addition of the class concept from object oriented modeling, it was found that the standard could be adapted to provide two principles of referencing of architecture elements. The first principle can expresses the generic structures of processes, organs and parts within the system, and the second principle can express specific configurations of a system. There is a need for further testing of using these two principles of referencing within a an architecture-centric development process, but it is argued that there is good reason to believe that providing a more stringent means of referencing core design elements will provide stakeholders with a better foundation for managing design information in regards to the architecture of complex production systems or families of production systems.

Part 3 of the ISO/IEC 81346 standard and the sector specific RDS for power plants also demonstrate how a reference designation system for design elements can be expanded to include referencing of signals and interfaces (terminals). The architecture description includes an interface diagram (TSV-2) intended for modeling of these very things, and it would therefore also make sense to expand the RDS presented here to include referencing of signals and interfaces. The TSV-2 has been prepared with this in mind

Part 6 Conclusion

The purpose of this research has been to investigate the phenomenon of production system architecture. Part 6 of the dissertation concludes on this research, summarizes the research contributions and offers perspectives for the use of the research within academia and industry.

34 Conclusion

The topic of this research is production system architecture and the description thereof. The research introduces contributions in the form a conceptual framework for understanding the architecture phenomenon; a conceptual model for architecture descriptions; and supporting tools for modeling of architecture and referencing of key architecture elements.

The theoretical goals of this research have been to contribute to the definition and understanding of the architecture phenomenon within the field of architecture-centric design, and to contribute to the operational description of production system architecture and handling of architecture related information.

The practical goals of this research have been to conduct the research with an industrial setting; to contribute to architecture-centric design of production systems within industry; to bridge the gap between academic research and industry by relating the research to industrial standards; and to provide an input for the consolidation of architecture and platform related research results within the research group.

This section concludes the dissertation by describing the research findings in relation to the research questions and summarizing the theoretical and practical contributions of the research. The three research questions of this research were presented in section 4.4. The research questions have been answered by parts two through five of this dissertation.

34.1 Research question 1

> ***RQ 1*** *What is production system architecture and how does the concept relate to existing theories of architecture within design of products and production systems?*
>
> Supporting questions
>
> - *What phenomena are described by production system architecture?*
> - *How does production system architecture relate to existing concepts of architecture used in the scientific community and industry?*
> - *What levels of production system architecture can be defined?*

Answer

Research question one has been answered in Part 2 of this dissertation.

The architecture phenomenon was investigated both through a review of literature from the scientific community and industry, and through observations made in case projects. It was found that there is a great difference in perception of what constitutes architecture and what sub-phenomena are included. The very basic nature of architecture is not commonly agreed upon, and it is variously defined as either an inherent aspect of a system or as a description of aspects of a system.

This research has found architecture to be an inherent aspect of a system and clarifies that there is a difference between architecture and its description. A production system can be said to have architecture or exhibit architecture, and architecture can be expressed by means of an architecture description. Production system architecture is therefore defined as:

> Fundamental concepts or properties of a production system embodied in its elements, relationships, and in the principles of the system's design and evolution that address the requirements and constraints from its intended applications.

Production system architecture is found to be a layered phenomenon that encompasses both design and application related phenomena of the system throughout its life-cycle, and the relation between the system design and the applications and roles of the system. A contribution is made in the form of a conceptual framework for the architecture phenomenon that is intended to aid in the understanding of the phenomenon (see Figure 79).

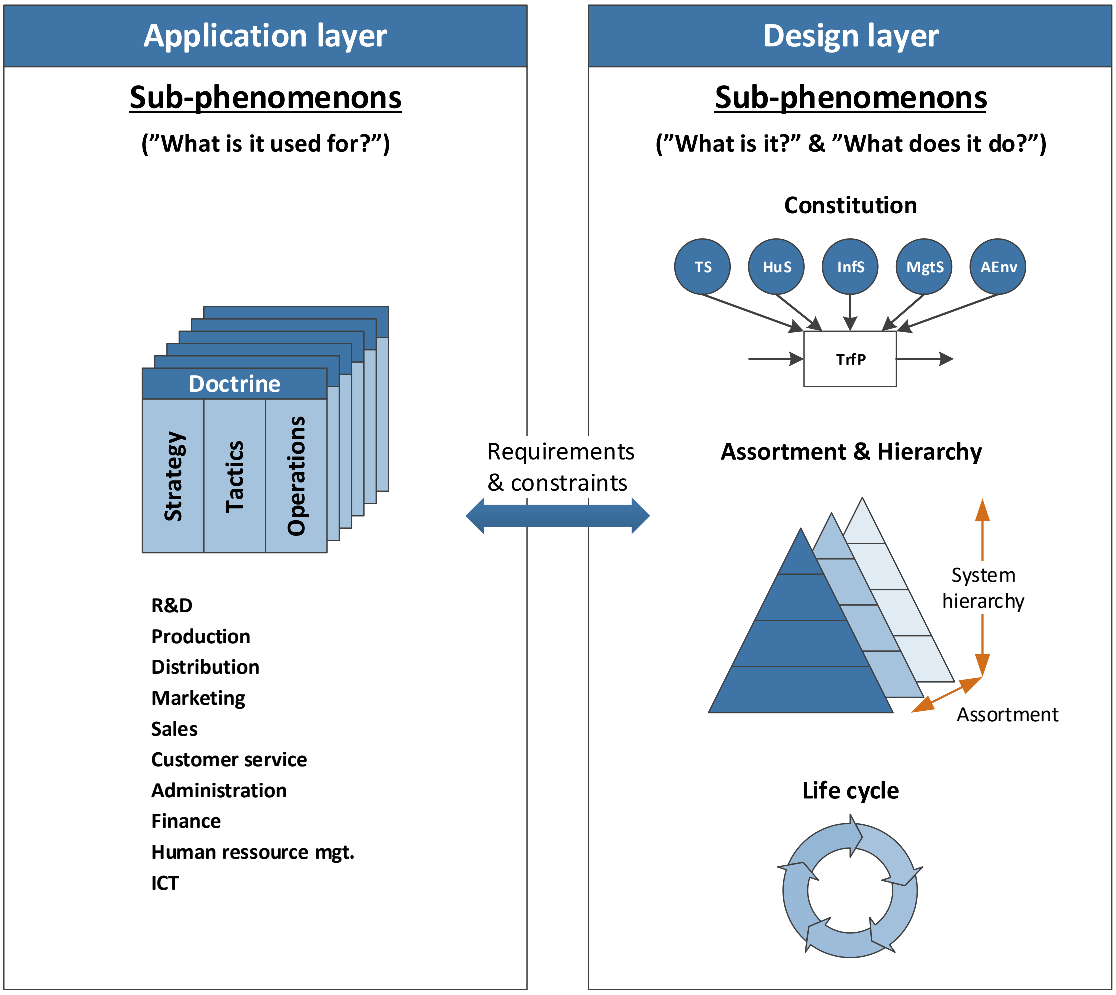


Figure 79 - Conceptual framework for the production system architecture phenomenon (same as Figure 12)

The framework describes two layers of the phenomenon, the Application layer and the Design layer:

Design layer: The architecture design layer describes phenomena associated with the constituent design of the production system. In other words the design layer of the system architecture can be said to answer the questions "What is it?" and "What does it do?" The design layer is found to be constituted by three sub-phenomena:

- Constitution: Describes the constituent design of the production system.
- Assortment & Hierarchy: Describes the variety of the systems and the recursive design phenomenon whereby systems are constituted by other systems.
- Life-cycle: Describes the life-cycle differences in the system design incl. the changeability of the system.

The three phenomena describe the system design, its relation to other systems, and its changeability in the life-cycle.

Application layer: A production system has several intended or unintended applications or roles to fulfill within the company, only one of which is the application of the system for production of products. The architecture application layer describes the phenomena related to the applications or roles the system fulfills. The application layer is found to be constituted by the four sub-phenomena Doctrine, Strategy, Tactics and operations, which can be found within different company functions. This means that the system's applications and roles must be defined in relation to the company's doctrines, strategies, tactics and operation phenomena e.g. the system's application in the product strategy or technology development strategy. The application phenomena provide the requirements and constraints which determine the principles of the systems organization and design as described in the architecture definition. In other words phenomena in the application layer can be said to answer the question "What is it used for?" The answer to this question provides the input for the design.

The connection between the application layer and the design layer represents the part of the architecture phenomenon that explains the connection between a system's design and its applications. The common definitions of architecture that have been encountered in literature often regard architecture as an expression of a system's core design characteristics defined by its elements, structures and the mapping between them. This is also covered by the provided definition as part of the design layer of the architecture phenomenon. However further study of the perception of architecture within industry associations, standards, and most importantly the case projects, have shown that the architecture phenomenon is also related to the intended applications or roles of the system and how these are fulfilled.

The provided conceptual framework can be said to expand on the existing concepts within product design and systems engineering, by offering a broader view of the architecture phenomenon. It could be summed up by saying that *architecture as a phenomenon is not just an expression of the core design of a system, but is also an expression of the link between system design and application.* The architecture of a system therefore not only provides an answer to how the system is designed, it also provides an answer to how the system design relates to the application of the system e.g. how does the design address the product strategy, technology strategy, financing tactics or plans in the production operation, or conversely, how are the strategies or operations constrained by the system design.

Some of the implications of this perception of the phenomenon lie in the description of the architecture of systems where the design is very unknown, as is the case in the early stages of design and for production system families, where some system's may not be built until years in the future. At a time when the design of the system is yet unknown or the system is not much more than a black box, the architecture of the system may still express how the future system will fit the intended applications expressed by requirements and constraints on the design or principles for the system's design and evolution. This perception of the architecture phenomenon reflects the way in which stakeholders treat the system at a time when the design is still unknown or very uncertain.

Throughout the research in industry it has been observed that there is a large range in the perception of production system architecture covering many different design or application related phenomena. It is believed that viewing production system architecture in the context of the conceptual framework will enable system stakeholders to better understand what phenomena are included in their perception. This should aid companies engaging in or transitioning to architecture-centric design in their internal communication and their treatment of the phenomenon in their organization and work processes. It should be noted however that the suggested conceptual framework needs further development. There is a need to expand upon the understanding of the application layer and the link between application and design.

34.2 Research question 2

RQ 2 *How can production system architecture be described, and what are the relevant elements and phenomena to describe in order to best support decision making on the design by stakeholders from different disciplines?*

Supporting questions

- *What stakeholders and stakeholder concerns should be addressed by a production system architecture description?*
- *What constituent elements and relations of a production system should be modeled as part of a production system architecture description?*
- *How can the architecture of a production system be modeled visually, including phenomena of scalability, interchangeability and flexibility of production systems be modeled?*

Answer

Research question two has been answered in Part 3 and Part 4 of this dissertation.

The description of production system architecture and system architecture in general has been investigated both by study of industrial standards and through case projects in industry. Contributions have been made in the form of a conceptual model of architecture descriptions, and a reference architecture framework for production systems, which includes viewpoints and model kinds for operational modeling of production system architecture.

The research into architecture description has found that the architecture of one or more production systems can be described by means of a so-called Architecture Description, which is a work product of an architecting process. A conceptual model for architecture descriptions have been formulated based on the ISO/IEC/IEEE 42010 standard (see Figure 80). The conceptual model ads a framework element to the concept as it is defined in the standard as part of a recognition that all architecture descriptions can be said to have a framework, which is specific to that architecture description alone.

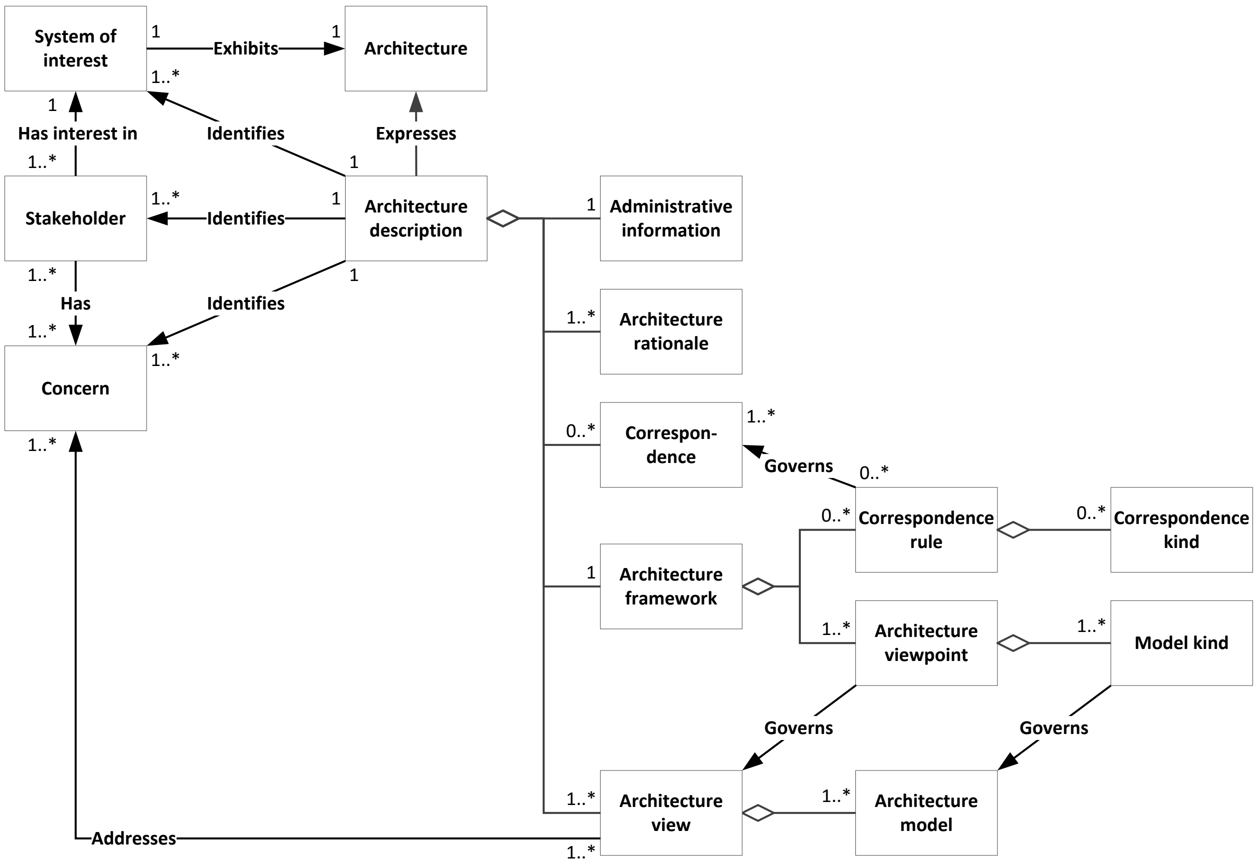


Figure 80 - Conceptual model for architecture descriptions (same as Figure 27)

An architecture description is used to express the architecture of a system-of-interest. It identifies the system-of-interest, the stakeholders having an interest in the system, and the concerns held by these stakeholders. The key elements of architecture descriptions are the architecture views which consist of different models helping to frame specific sets of stakeholder concerns by modeling different aspects of the system; and the architecture framework which specifies the viewpoints governing the views and the correspondence rules that govern the correspondence in the description.

The frameworks of specific architecture descriptions can be based on sector/domain specific reference architecture frameworks. Reference architecture frameworks consolidate specific viewpoints, model kinds, correspondence rules, and correspondence kinds used within specific sectors e.g. the defense industry.

For the description of production system architecture a contribution has been made in the form of a reference framework based on which architecture descriptions for production systems can be generated. The presented version of the reference framework i.e. the Production System Architecture Framework

(PSAF) contains two viewpoints that frame some of the concerns of stakeholders but not all. The two viewpoints are the Production capability viewpoint, which frames concerns relating to the production task of the system, and the Technical system viewpoint, which frames concerns regarding the design of the technical system within the production system. Each viewpoint contains model kinds for the description of different aspects of a production system architecture including phenomena such as scalability, changeability and flexibility. Despite the number of included model kinds, it is concluded that more viewpoints and model kinds will be needed in order to provide a comprehensive description of a production system architecture that addresses all relevant stakeholder concerns.

It has also been observed that depending on the need for architecture description, a satisfactory description can be achieved by use of a subset of the model kinds. Meaning that the reference framework is meant to be adapted to form a framework specific to the particular architecture description of a production system. A framework for a specific production system architecture description will therefore most likely use a subset of the model kinds to frame the concerns of its particular group of stakeholders.

The PSAF has been developed based on the identified stakeholders and concerns of relevance for production system architecture. The types of stakeholders of relevance for the description of production system architecture have been found to be very wide. A production system has many different applications or roles within a company, and it follows that the architecture of the system is of relevance for stakeholders throughout the company. Because the architecture not only describes the constituent design of a production system, but also explains the relation between the system application and the design, the architecture description must be able to communicate aspects of the architecture to stakeholders from vastly different backgrounds. The architecture description therefore must include descriptive means that can be used in communication between these stakeholders. The stakeholder concerns of interest for the description of production system architecture have been investigated in section 20. It was found that the concerns could be found in both the direct life-cycle processes of the system of interest and in processes not related to the life-cycle. Key concerns of interest have been described in the section.

In regards to what should be modeled as part of a description of architecture, it is found that it is not simply a matter of what constituent elements and relations of a system should be modeled, but also their relations/relationships to the applications and roles within the company. The architecture not only expresses the constituent design of the production system, but also expresses the connection between the applications and roles of the system and the design of the system. Therefore the architecture description must model not only model the design of the system, but also the applications or their link to the system design. In regard to modeling of the design of the production system, it must be said that the architecture models the production system, which is defined as a transformation system. The elements main that must be modeled are therefore found in the model of a transformation system. This includes the process, the operators and their relations. In this research project there has been a particular emphasis on the modeling of the technical system and the production process. A viewpoint has been defined which in particular focuses on the technical system.

As it stands the Production System Architecture Framework developed in this research project is still in need of further development. There is a need for further description of the system applications and their link to the system design, and there is a need for description of the other elements of the production system than the Technical System. This task has unfortunately proven too large for this project to encompass, and must be handled as part of future research into description of production system architecture.

34.3 Research question 3

RQ 3 *How can the ISO/IEC 81346 standard series be applied to support exchange and processing of architecture information within and between stakeholder domains and tool in the production system life cycle as part of the description of production system architecture?*

Supporting questions

- *What production system elements and structures are relevant to model to increase communication of elements, structures and phenomena between stakeholders, design tools and IT systems as part of the description of system architecture?*
- *How can the ISO/IEC 81346 standard series be applied for modeling relations between system elements such as type commonality and functional allocation?*

Answer

Research question one has been answered in Part 5 of this dissertation.

Research question three has been investigated through analysis of ISO/IEC 81346 and sector specific reference designation systems based on the series; experiences from consultancy work with implementation of ISO/IEC 81346 in companies designing and building large processing plants; and my involvement in the development of a sector specific reference designation system for the Danish building industry. The RDS has also been tested in a procurement project in the primary case company.

Investigation of the standard and sector specific applications of the standard found that there were several limitations in the basic concepts and capabilities of the standard. The RDS does not directly allow for referencing of generic structures and variants of system elements, since the standard is aimed at describing specific system configurations, i.e. it is intended to be applied in the referencing of individual systems and not families of systems. The standard is also not intended to be used to reference potential future additions to the systems. It has also been found that the basic concept of aspects require a more firm definition, or delimitation to what they express, if they are to be used to reference objects in an architecture. It must be explicitly defined what objects are structured under an aspect. Most importantly the standard does not model class structures, which are necessary to express the generic structures of the production system and variety of design elements. In order to apply the standard it is therefore necessary to modify it to serve the purpose of referencing within an architecture description. In order to apply the standard the following key changes have been made:

- the addition of the class concept
- the addition of class structures for description of generic structures of the system and element variety

- The addition of a coding mechanism for identifying variants of elements as part of the generic structures.

A contribution to support exchange and processing of architecture information has been made by adapting the ISO/IEC 81346 for identification and referencing of key production system elements. Specifically the standard series has formed the basis of the development of a correspondence kind which is included in the Production System Architecture Framework (PSAF). This correspondence kind can be used in referencing of key system elements across models of an architecture description and also in other documents, models or IT-systems not a part of the architecture description. The correspondence kind is a so-called reference Designation System, which is a structuring and coding system that can be used to assign reference designations (a kind of ID's) to key elements of the production system based on the compositional relations of the element. Within the PSAF the reference designation system is used to provide reference designations to the processes, organs and parts of the production system, so that these can be explicitly identified and referenced within architecture descriptions. This will support the exchange of information for complex systems by eliminating uncertainty as to what element the exchanged information is concerning.

The developed reference designation system allows for referencing of generic system structures and specific configurations of system structures. This means that the system can both reference the design elements defined in the system's generic structures and the instances of those elements that exist in a specific system configuration (see Figure 81). As an example the generic part structure of production system may express that the system can contain a Press, which is used to perform a specific Technical Process. The specific part structure of the system may express that the system contains three instances of the Press. The reference designation system enables referencing of both the press in the generic part structure, which is the subject of design activities, and the three instances of the press that exist in the final system, which are the subject of activities such as procurement, build, utilization, service, etc. If there exists variants of the press (e.g. different models), then the reference designation system also enables referencing of these.

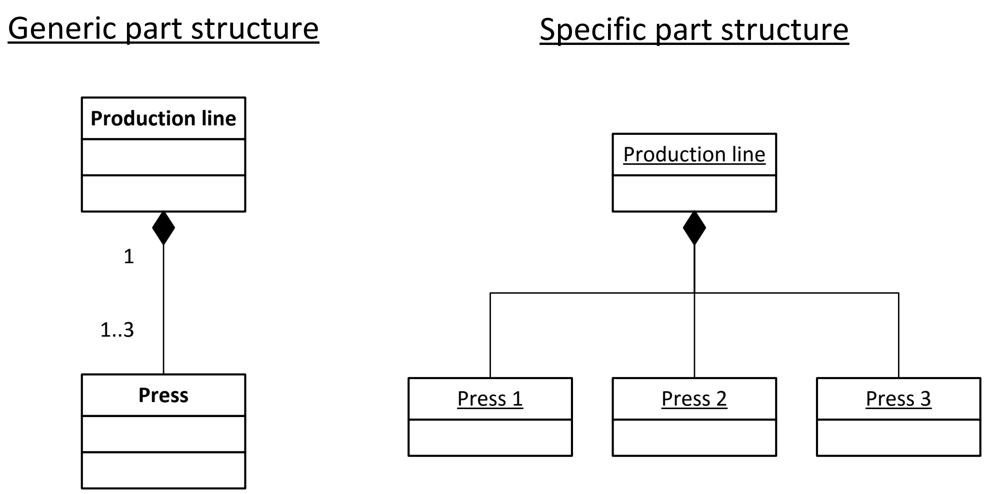


Figure 81 - The system structure can be expressed both through a generic structure showing the principle of the design and a specific structure showing the specific makeup of the system (same as Figure 73).

It has been found that only some relations between elements of the production system can be expressed by use of a reference designation system. The structuring principles of ISO/IEC 81346 only allow for compositional relations as the basis of the structuring of objects. Because the aspects of the suggested reference designation system are tied directly to domain (process domain, organ domain and prat domain), the system does not allow for multiple aspects to be used in the structuring of elements and forming of reference designations. This means that relations such as functional allocation cannot be directly expressed by a reference designation. These kinds of relations can only be expressed in documents, models, IT-systems, etc. However the use of reference designations does allow for explicit identification of the elements that are whose relation is expressed in these contexts. Other relations such as type commonality can be expressed by the use of either object classes or meta-classes.

The application of the standard series currently only identifies three types of system elements (processes, organs and parts). While these elements are among the most important to identify and reference for a production system architecture, many other elements modeled in a production system architecture description are also of importance and should be referenced across models. Future research should determine what other elements of an architecture description should be covered by a reference designation system, e.g. interfaces and interactions between elements.

34.4 Validation of results

The majority of the conclusions and contributions of this research are built upon cases from within the primary case company that have a starting point in the observation of a practical problem in relation to the design and development of production systems. Although these cases have been greatly varied and had a broad reach within the organization, it is a concern of this research that the observations have not been confirmed in other companies, but are only supplemented by the practical experience of the researcher and the research group. This causes a problem with validating the results of the research as mentioned in the start of this dissertation. I have sought to counter this issue by:

- Seeking confirmation in literature from both the scientific community and industry.
- Relating the research to industrial standards.
- Building many of the included modeling formalisms on known and proven modeling formalisms from the product design field.
- Supplementing the action research in the case company with work experience from outside the research project.
- Using a variety of methodologies as the basis of the research, including observation, literature study, interviews, workshop and direct participation.
- Taking a broad approach to the investigation of the architecture phenomenon which has not just been limited to modeling of architecture.

I would conclude that while there is certainly a need for further validation of the results, this is a common trait of research within this field which is heavily reliant on long term observation and testing within companies. It is my belief that the research conducted in this project has had a sufficient basis in both theory and praxis to accept the results as valid within the context of architecture-centric design of production systems for case companies facing the same design and development task as the Grundfos Technology Center.

35 Impact of research

This section presents observations on the research impact for industry and the scientific community.

35.1 Impact for industry

It was stated in the beginning of this research that architecture-centric design of production systems is one of the means by which industry can address the challenges they face in maintaining production within the Danish and European manufacturing industry. This research has contributed to the vocabulary within architecture-centric design, the understanding of the architecture phenomenon and the operational description of production system architecture. It is believed that the suggested perception of architecture and the reference architecture framework for production systems can aid companies in their transition to architecture-centric design of production systems, or that it can strengthen their existing practices. This is seen as a valuable aid in achieving the advantages of architecture-centric design and platform based development of production systems. Using a common framework for description of production system architecture also has the has the potential to aid the collaboration between companies and their suppliers of production equipment.

35.2 Impact for the design research community

In general the work in this research project has convinced me that the design research community in Europe, and certainly within the Danish academic environment, should consider utilizing other means of dissemination to industry for their methods, models and tools. A few examples are given here.

<u>Input for standards</u>

International standards provide companies with a more formal and consistent means of applying results from the research community than the current scientific publications allow. Standards have much more clear requirements for conformance, which is especially important when collaborating companies seek to apply similar means of working and communicating. I believe that the design research community should take a more active role in the development of standards for industry in order to provide industry with a more rigorous specification of methodologies and concepts.

<u>Handbook of architecting</u>

It is believed that an equivalent reference framework could be established for product architecture descriptions, a so-called Product Architecture Framework (PAF). The product architecture group at DTU-MEK and the design community has a body of research for product architecture and product platforms, with many modeling tools used for description of product architecture and platforms. Consolidating these contributions and utilizing a common means of describing them within a framework for description of product architecture would allow for easier communication of contributions both within the design research community and to industry. It would also allow for more easy comparison of research contributions and descriptions of the novelty of research contributions. The subject of dissemination of the research design community´s research is an issue for almost all research groups. It could serve the Design Society (www.designsociety.org) well to try and formalize the available knowledge, models, processes etc. in a framework for descriptions, and possibly also in a handbook of design similar to that of INCOSE. If it is possible for the Systems engineering field to describe a formal methodology as expressed in a handbook and international standards, then surely something similar could be attempted in the design research community. It might take some years, but it would surely be worth it in terms of getting the industry to more readily adopt the developed methods and tools by providing a more formalized reference and explanation for their usability and use.

Certification

Formalizing some of the research contributions within the design research community also raises the possibility of establishing certification in relation to architectural design similar to e.g. the Systems engineering certification from INCOSE, or Six Sigma and Lean certification. The existence of such certifications has had a great impact on the dissemination of systems engineering practices and in the adoption of methods within industry. It raises the possibility for the research community to certify members as Systems Architects and have companies across different countries understand what this certification signifies. This kind of formalization does not necessarily have to limit designers and systems architects. INCOSE offers a good example of a definition of Systems Engineering, with a wide range of freedom, but still offers some common frame of reference for the systems engineering discipline. I believe this perspective ought to be investigated more by the design research community.

35.3 Impact for research group

Within the Product Architecture research group at the Department of Mechanical Engineering at DTU several modeling kinds have been developed e.g. Product family Master Plan, Interface diagrams, Generic Production Flows, etc. It could serve the group well to consolidate the results in a framework for description of product architecture, with a clear and explicit description of the viewpoints, rationales, stakeholders and concerns. This could serve as a collection of the research results, and would provide a more complete package to industry of the results generated in the research group. Some semblance of this has already taken place in the project “Radikal forenkling via design” conducted in collaboration with the Danish Industry Foundation (Industriens Fond, 2012). The project included an attempt to describe to industry the approach applied in the architecture work of the research group. It would serve the group well to formalize this even more, with clear descriptions of the viewpoints used, the underlying rationales of the viewpoints, and the modeling kinds used. This could either be done in based on the conceptual model for architecture description presented in this research or the conceptual model of ISO/IEC/IEEE 42010. This would provide the group with a more clear description of the research contributions internally, allow for consolidation of results from the different researchers, and it would provide a common package to use in regards to dissemination in the education of engineering students, and dissemination to engineers in industry.

36 References

Alabastro, M.S., Beckmann, G., Gifford, G., Massey, A.P. and Wallace, W.A., 1995. The use of visual modeling in designing a manufacturing process for advanced composite structures. *IEEE Transactions on Engineering Management*, 42(3), pp.233–242.

Andreasen, M.M., 1980. *Syntesemetoder på systemgrundlag : bidrag til en konstruktionsteori*. Ph.D. Lund Institute of Technology.

Andreasen, M.M., 2007. How to spell a product? Lecture at Ilmenau University of Technology, Ilmenau, Germany, 5 july 2007. (Unpublished).

Andreasen, M.M., Hansen, C.T. and Mortensen, N.H., 1995. On structure and structuring. In: H. Meerkamm, ed., *6. Symposium Fertigungsgerechtes Konstruieren*. Erlangen, Germany, 19-20 October 1995. Erlangen: University of Erlangen-Nuremberg, pp.1–15.

Andreasen, M.M., Hansen, C.T. and Mortensen, N.H., 1996. The structuring of products and product programmes. In: M. Tichem, M.M. Andreasen and K.J. MacCallum, eds., *2nd WDK Workshop on Product Structuring*. Delft, The Netherlands, 3-4 June 1996. Delft: Delft University of Technology, pp.15–43.

Andreasen, M.M., Howard, T.J. and Bruun, H.P.L., 2014. Domain theory, its models and concepts. In: A. Chakrabarti and L.T.M. Blessing, eds., *2014. An anthology of theories and models of design : philosophy, approaches and empirical explorations*. London: Springer-Verlag, pp.173 – 195.

Andreasen, M.M., Mortensen, N.H. and Harlou, U., 2004. Multi product development : new models and concepts. In: H. Meerkamm, ed., *15th Symposium on Design for X*. Neukirchen/Erlangen, Germany, 14-15 October 2004. Erlangen: Friedrich-Alexander-University, pp.75–86.

Balslev, H., 2010. *Appliance of reference designations on machinery in accordance with ISO/IEC/EN 81346*. [pdf] Knowledge Center of Reference Designations ISO/IEC 81346. Available at: <http://81346.com/english/?page_id=186>.

Berglund, F., Bergsjö, D., Högman, U. and Khadke, K., 2008. Platform strategies for a supplier in the aircraft engine industry. In: ASME, ed., *2008 ASME International Design Engineering Technical Conferences & Computers and Information Engineering Conference.* New York USA, 3-6 August 2008. New York: American Society of Mechanical Engineers, p.Vol 4 (55–66).

Booch, G., Maksimchuk, R.A., Engle, M.W., Young, B.J., Conallen, J. and Houston, K.A., 2007. *Object-oriented analysis and design with applications*. 3rd ed. Upper Saddle River: Addison-Wesley.

Bruun, H.P.L. and Mortensen, N.H., 2012. Modelling and using product architectures in mechatronic product development. In: P.A. Hansen, J. Rasmussen, K.A. Jørgensen and C. Tollestrup, eds., *NordDesign 2012*. Aalborg, Denmark, 22-24 August 2012. Aalborg: Aalborg University.

Bruun, H.P.L., Mortensen, N.H. and Harlou, U., 2014. Interface diagram: Design tool for supporting the development of modularity in complex product systems. *Concurrent Engineering : research and Applications*, 22(1), pp.62–76.

Buch-Hansen, H. and Nielsen, P., 2005. *Kritisk realisme*. Frederiksberg: Roskilde Universitetsforlag.

Buur, J., 1990. *A theoretical approach to mechatronics design*. Ph.D. Technical University of Denmark.

Coad, P. and Yourdon, E., 1991. *Object-oriented analysis*. 2nd ed. Englewood Cliffs: Yourdon Press.

Crawley, E., de weck, O., Eppinger, S.D., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. and Whitney, D., 2012. Engineering systems monograph : the influence of architecture in engineering systems. In: *Engineering Systems Symposium*. Cambridge, USA, 29-31 March 2004. Cambridge: MIT Engineering Systems Division.

cuneco, 2012. *CCS kodningsregler - Kodningsregler for klassifikation og identifikation af objekter - Foreløbig udgave til offentlig høring 2. udgave*. [pdf] cuneco - center for produktivitet i byggeriet. Available at: <http://cuneco.dk/ccs-kodestruktur#3>.

cuneco, 2014. *CCS Identifikation (R3)*. [pdf] cuneco - center for produktivitet i byggeriet. Available at: <http://cuneco.dk/vaerktoejer>.

DI, 2013. *Sådan ligger landet : globaliseringsredegørelse 2013*. Copenhagen: Confederation of Danish Industry (Dansk Industri).

Eder, W.E., 2011. Theory of Technical Systems - Relationships to Engineering Sciences. In: *Canadian Engineering Education Association Conference 2011*. St. Johns, Canada, 6-8 June 2011. St. Johns: Memorial University St. Johns.

Eder, W.E. and Hosnedl, S., 2007a. *Design engineering : a manual for enhanced creativity*. Boca Raton: CRC Press.

Eder, W.E. and Hosnedl, S., 2007b. Transformation systems : revisited. In: J.-C. Bocquet, ed., *16th International Conference on Engineering Design*. Paris, France, 28-31 August 2007: Design Society.

ElMaraghy, H.A., 2005. Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*, 17(4), pp.261 – 276.

Emery, D. and Hilliard, R., 2009. Every architecture description needs a framework : expressing architecture frameworks using ISO/IEC 42010. In: IEEE, ed., *2009 Joint Working IEEE/IFIP Conference on Software Architecture & 3rd European Conference on Software Architecture*. Cambridge, United Kingdom, 14-17 September 2009. New jersey: IEEE Computer Society, pp.31–40.

Ericsson, A. and Erixon, G., 1999. *Controlling design variants : modular product platforms*. Dearborn: Society of Manufacturing Engineers.

García, R.G. and Gelle, E., 2006. Applying and Adapting the IEC 61346 Standard to Industrial Automation Applications. *IEEE Transactions on Industrial Informatics*, 2(3), pp.185–191.

German Institute for Standardization, 2003. *DIN 8580:2003-09 Fertigungsverfahren - Begriffe, Einteilung*. Berlin: German Institute for Standardization.

Golden, B., 2013. *A Unified Formalism for Complex Systems Architecture*. Ph.D. École Polytechnique.

Hansen, C.T. and Andreasen, M.M., 2002. Two approaches to synthesis based on the domain theory. In: A. Chakrabarti, ed., *Engineering design synthesis : understanding, approaches and tools*. London: Springer-Verlag, pp.93–108.

Harlou, U., 2006. *Developing product families based on architectures : contribution to a theory of product families*. Ph.D. Technical University of Denmark.

Harry, M.J. and Schroeder, R., 2000. *Six Sigma : the breakthrough management strategy revolutionizing the world's top corporations*. New York: Currency.

Hilliard, R., 2014. *Survey of Architecture Frameworks*. [online] Available at: <http://www.iso-architecture.org/42010/afs/frameworks-table.html> [Accessed 16 May 2014].

Hilliard, R., Malavolta, I., Muccini, H. and Pelliccione, P., 2012. On the composition and reuse of viewpoints across architecture frameworks. In: *2012 Joint Working Conference on Software Architecture & 6th European Conference on Software Architecture*. Helsinki, Finland, 20-24 August 2012. New Jersey: IEEE Computer Society, pp.131–140.

Hubka, V., 1967. Der grundlegende algorithmus für die lösung von konstruktionsaufgaben. In: *XII. Internationales Wissenschaftliches Kolloquium der Technischen Hochschule Ilmenau*. Ilmenau: Ilmenau University of Technology, pp.69–74.

Hubka, V. and Eder, W.E., 1988. *Theory of technical systems: a total concept theory for engineering design*. Berlin and New York: Springer-Verlag.

Högman, U., 2011. *Processes and platforms aligned with technology development : the perspective of a supplier in the aerospace industry*. PhD. Chalmers University of Technology.

INCOSE, 1998. *INCOSE SE Terms Glossary, Version 0*. Seattle, USA: [pdf] International Council on Systems Engineering (INCOSE). Available at: <https://www.incose.org/>.

INCOSE, 2011. *Systems engineering handbook : a guide for system life cycle processes and activities : version 3.2.2*. San Diego: International Council on Systems Engineering (INCOSE).

Industriens Fond, 2012. *Radikal Forenkling via Design*. Kgs. Lyngby: DTU Mekanik.

ISO, 2012. *ISO/TS 81346-3:2012(E) Industrial systems, installations and equipment and industrial products : structuring principles and reference designations : part 3 : application rules for a reference designation system*. Geneva: ISO.

ISO and IEC, 2007. *ISO/IEC 42010:2007(E) Systems and software engineering : recommended practice for architectural description of software-intensive systems*. Geneva: ISO.

ISO and IEC, 2008. *ISO/IEC 15288:2008(E) Systems and software engineering : system life cycle processes*. Geneva: ISO.

ISO and IEC, 2009a. *ISO/IEC 81346-1:2009(E) Industrial systems, installations and equipment and industrial products : structuring principles and reference designations : part 1 : basic rules*. Geneva: ISO.

ISO and IEC, 2009b. *ISO/IEC 81346-2:2009(E) Industrial systems, installations and equipment and industrial products : structuring principles and reference designations : part 2 : classification of objects and codes for classes*. Geneva: ISO.

ISO, IEC and IEEE, 2011. *ISO/IEC/IEEE 42010:2011(E) Systems and software engineering : architecture description*. Geneva: ISO.

Jiao, J., Simpson, T.W. and Siddique, Z., 2007. Product family design and platform-based product development : a state-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1), pp.5–29.

Jiao, J., Tseng, M.M., Ma, Q. and Zou, Y., 2000. Generic bill-of-materials-and-operations for high-variety production management. *Concurrent Engineering*, 8(4), pp.297–321.

Jiao, J., Zhang, L. and Pokharel, S., 2003. Process platform planning for mass customization. In: *2nd Interdisciplinary World Congress on Mass Customization and Personalization*. Munich, Germany, 6-8 October 2003. Munich: Technical University of Munich, pp.1–35.

Johansen, J., Madsen, O., Jensen, H.V. and Vestergaard, A., 2010. *Manufacturing 2025 : five future scenarios for Danish manufacturing companies*. Aalborg: [pdf] Aalborg University. Available at: <http://www.manufuture.dk/>, pp.1–36.

Jørgensen, K.A., 1992. *Videnskabelige arbejdsparadigmer*. [working paper] Aalborg: Aalborg University. Available at: <http://www.kaj.person.aau.dk/Research/Publications/>.

Klir, J. and Valach, M., 1967. *Cybernetic modelling*. London: Iliffe Books Ltd.

Krafcik, J.F., 1988. Triumph of the lean production system. *MIT Sloan Management Review*, 30(1), pp.41–52.

Kvist, M., 2010. *Product Family Assessment*. Ph.D. Technical University of Denmark.

Levandowski, C.E., Corin-Stig, D., Bergsjö, D., Forslund, A., Högman, U., Söderberg, R. and Johannesson, H., 2012. An integrated approach to technology platform and product platform development. *Concurrent Engineering*, 21(1), pp.65–83.

Manufuture-DK, 2013. *Manufuture-DK - Vision for manufuture*. [online] Available at: <http://www.en.manufuture.dk/about-manufuture/vision-for-manufuture/>.

Manufuture-EU, 2013. *Manufuture -EU: Objectives*. [online] Available at: <http://www.manufuture.org/manufacturing/?page_id=734 >.

Martin, M. V. and Ishii, K., 2002. Design for variety : developing standardized and modularized product platform architectures. *Research in Engineering Design*, 13(4), pp.213–235.

Merriam-Webster, n.d. *Class*. [online] Merriam-Webster.com. Available at: <http://www.merriam-webster.com/dictionary/class>.

Meyer, M.H. and Lehnerd, A.P., 1997. *The power of product platforms : building value and cost leadership*. New York: The Free Press.

Ministère de la Défense, 2005. *Guide AGATE*. [pdf] Ministère de la Défense. Available at: <http://www.achats.defense.gouv.fr/article33349>.

Minsky, M.L., 1968. Matter, Minds and Models. In: M.L. Minsky, ed., *Semantic Information Processing*. Cambrige: MIT Press, pp.425–432.

Mortensen, N.H., 1999. *Design modelling in a designer's workbench : contribution to a design language*. Ph.D. Technical University of Denmark.

Mortensen, N.H. and Andreasen, M.M., 1996. Designing in an interplay with a product model : explained by design units. In: *1996 International Symposium on the Tools and Methods for Concurrent Engineering*. Budapest, Hungary, 29-31 May 1996. Budapest: Technical University of Budapest, pp.100–114.

Mortensen, N.H., Hansen, C.L., Hvam, L. and Andreasen, M.M., 2011. Proactive modeling of market, product and production architectures. In: S. Culley, B. Hicks, T. McAloone, T. Howard and J. Clarkson, eds., *18th International Conference on Engineering Design*. Kgs. Lyngby, Denmark, 15-19 August 2011.: Design Society, pp.133 – 144.

Mortensen, N.H., Pedersen, R., Kvist, M. and Hvam, L., 2008a. Modelling and visualising modular product architectures for mass customisation. *International Journal of Mass Customization*, 2(3/4), pp.216–239.

Mortensen, N.H., Pedersen, R., Nielsen, O.F., Harlou, U., Bøgh, M.H., Høgh, P.T. and Hvam, L., 2008b. Implementing a product platform in 35 man-days : the visual thinking approach. *International Journal of Mass Customisation*, 2(3/4), pp.240–263.

NASA, 2013. *NASA Procedural Requirements 7123.1B - Appendix E - Technology Readiness Levels*. [online] NASA Online Directives Information System. Available at: <http://nodis3.gsfc.nasa.gov/>.

Olesen, J., 1992. *Concurrent development in manufacturing : based on dispositional mechanisms*. Ph.D. Technical University of Denmark.

OPEN Process Framework Repository Organization, 2009. *OPEN Process Framework (OPF) - Architecture*. [online] Available at: <http://www.opfro.org/index.html?Components/WorkProducts/ArchitectureSet/Architectures/Architectures.html~Contents> [Accessed 16 May 2014].

Pedersen, R., 2010. *Product platform modelling : Contributions to the discipline of visual product platform modelling*. Ph.D. Technical University of Denmark.

Phaal, R., Farrukh, C.J.P. and Probert, D.R., 2001. Characterisation of technology roadmaps : purpose and format. In: *Portland International Conference on Management of Engineering and Technology 2001*. Portland, USA, 29 July - 2 August 2001. New Jersey: IEEE, pp.367–374.

Pimmler, T.U. and Eppinger, S.D., 1994. Integration analysis of product decompositions. In: *ASME 1994 International Design Engineering Technical Conferences & 6th Design Theory and Methodology Conference.* New York, Minneapolis, USA, 11-14 September 1994: American Society of Mechanical Engineers, pp.343–351.

Pine II, B.J., 1993. *Mass customization: The new frontier in business competition*. Boston: Harvard Business School Press.

Rao, V., 2007. *Strategy, tactics, operations and doctrine : a decision-language tutorial*. [online] Available at: <http://www.ribbonfarm.com/2007/09/24/strategy-tactics/> [Accessed 19 May 2014].

Ravn, P.M., Guðlaugsson, T.V. and Mortensen, N.H., 2014. Visual modelling of pilot production to support decision making in production development. In: D. Marjanović, M. Štorga, N. Pavković and N. Bojčetić, eds., *Design 2014 - 13th International Design Conference*. Dubrovnik, Croatia, 19-22 Maj 2014: Design Society, pp.1179–1188.

Reichwein, A. and Paredis, C.J.J., 2011. Overview of architecture frameworks and modeling languages for model-based systems engineering. In: *ASME 2011 International Design Engineering Technical Conferences & 31st Computers and Information in Engineering Conference.* Washington, USA, 29-31 August 2011. New York: American Society of Mechanical Engineers, pp.1341–1349.

Sage, A.P., 1992. *Systems Engineering*. New York: Wiley-Interscience.

Salvador, F., Forza, C. and Rungtusanatham, M., 2002. Modularity, product variety, production volume, and component sourcing : theorizing beyond generic prescriptions. *Journal of Operations Management*, 20(5), pp.549–575.

Sanchez, R., 2004. Creating modular platforms for strategic flexibility. *Design Management Review*, 15(1), pp.58–67.

Sawhney, M.S., 1998. Leveraged high-variety strategies : from portfolio thinking to platform thinking. *Journal of the Academy of Marketing Science*, 26(1), pp.54–61.

Schierholt, K., 2001. Process configuration : combining the principles of product configuration and process planning. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 15(5), pp.411–424.

Skinner, W., 1969. Manufacturing : missing link in corporate strategy. *Harvard Business Review*, 47(3), pp.136–145.

Tjalve, E., 2003. *Systematic design of industrial products*. Kgs. Lyngby: Institute for Product Development.

Tripathy, A. and Eppinger, S.D., 2007. *A system architecture approach to global product development*. [working paper] Cambridge: MIT Sloan School of Management. Available at: <http://mitsloan.mit.edu/pdf/global-product.pdf>.

Tseng, M.M. and Jiao, J., 1998. Design for mass customization by developing product family archtecture. In: *ASME 1998 Design Engineering Technical Conferences & 3rd Design for Manufacturing Conference.* Atlanta, USA, 13-16 September 1998. New York: American Society of Mechanical Engineers.

Ulrich, K., 1995. The role of product architecture in the manufacturing firm. *Research Policy*, 24(3), pp.419 – 440.

Ulrich, K.T. and Eppinger, S.D., 2004. *Product design and development*. 3rd ed. Boston: McGraw-Hill/Irwin.

Westkämper, E., 2006. Digital manufacturing in the global era. In: P.F. Cunha and P.G. Maropoulos, eds., *2007. Digital Enterprise Technology: Perspectives and Future Challenges*. New York: Springer, pp.3–14.

Wiendahl, H.-P., ElMaraghy, H. a., Nyhuis, P., Zäh, M.F., Wiendahl, H.-H., Duffie, N. and Brieke, M., 2007. Changeable manufacturing - classification, design and operation. *CIRP Annals - Manufacturing Technology*, 56(2), pp.783–809.

Womack, J.P. and Jones, D.T., 2003. *Lean thinking : banish waste and create wealth in your corporation*. New York: Free Press.

Yassine, A.A. and Wissmann, L.A., 2007. The implications of product architecture on the firm. *Systems Engineering*, 10(2), pp.118–137.

Yu, J.S., Gonzalez-zugasti, J.P. and Otto, K.N., 1999. Product architecture definition based upon customer demands. *Journal of Mechanical Design*, 121(3), pp.329 – 335.

AAU, 2010. *Produktion 2025 – katalog for udviklings- og forskningstemaer*. [pdf] Aalborg: Aalborg University. Available at: <http://www.manufuture.dk/Aktiviteter/Projektkatalog/>.

DTU Mechanical Engineering
Section of Engineering Design and Product Development
Technical University of Denmark

Produktionstorvet, Bld. 426
DK- 2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 6263
Fax (+45) 4593 1577
www.mek.dtu.dk
ISBN: 978-87-7475-423-7

DCAMM
Danish Center for Applied Mathematics and Mechanics

Nils Koppels Allé, Bld. 404
DK-2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 4250
Fax (+45) 4593 1475
www.dcamm.dk
ISSN: 0903-1685