**NordDesign 2020**
**August 11-14, 2020**
**Kgs. Lyngby, Denmark**

# Conceptual model of modularisation

**Jarkko Pakkanen[1], Tero Juuti[1], Timo Lehtonen[1], Jari Peltonen[2], Rami Laine[2]**

[1]*Tampere University*
jarkko.pakkanen@tuni.fi
tero.juuti@tuni.fi
timo.lehtonen@tuni.fi
[2]*Cometa Solutions Oy*
jari.peltonen@cometasolutions.fi
rami.laine@cometasolutions.fi

## Abstract

Mass-producing companies of standard products need to become more agile because of the customisation trend. Capabilities to respond to customisation requirements is also required in the manufacturing industry, where production volumes can be small and products large, complex and expensive. The delivery chain for unique products puts a lot of pressure on companies, because almost the same product has to be invented over and over again in different projects. Product development tools are needed in developing capability to produce customisable products. Efficiently customisable and modular products enable many benefits in the supply chain if successfully implemented. For example, engineering resources may be freed from engineering work in delivery projects to develop new concepts. Few modularisation projects succeed in the first time. Designing a modular product architecture is difficult. In many cases, modularisation has been started on a purely technical basis, without consideration of the business environment and the type of design information that will allow the benefits to be realised. Due to a lack of understanding of the design information associated with the modular product architecture, it has not been well documented and the development of tools has been limited except a few consulting companies. Software commonly used by industry does not focus on modularisation. This research is part of a research project to develop software support for developing and managing modular products in different environments. We discuss problem definition, solution goals, and criteria definition for the modularisation support tool. It is difficult to produce and manage information for which there is no conceptual model. Therefore, the results of the paper illustrate what kind of design information is specific for a modular product. This is also described as a conceptual model, because without it, software supporting development work cannot be done.

***Keywords: Modularisation, modularity, conceptual model***

# 1 Introduction

Demand for customisation is one of the many megatrends (Deloitte, 2017). Mass-producing companies manufacturing standard products are forced to become more agile as the customisation trend spreads. Manufacturers operating in the project business, where the customer gets a unique product that meets their specific requirements, also have to develop their operating methods and products. The delivery chain for unique products and parts puts a lot of pressure on companies if product customisation is not intelligently designed. Methods and tools are needed to support designing these capabilities to respond to customisation trend.

Andreasen (2011) explains that the aim of modularisation is to reduce complexity in business operations while enabling customising the products from the customer's point of view so that similarities between the alternative building blocks, typically called module variants, exists. He continues stating that modularisation includes designing modular product architecture and modules. Fujimoto (2007) defines that in a modular product architecture, interactions between modules occur through defined interfaces. The importance of interface documentation (Stevens, 1998) and modular product structures (Hanna, Schwede, & Krause, 2018) has been presented in the context of systems engineering. In addition to these, model-based approaches to support modularisation have also been studied (Albers et al., 2019). Förg, Stocker, Kreimeyer, & Lienkamp (2014) and Holmqvist (2004) suggest that space reservations for modules and layout options must be considered in designing modular architecture. Product family based on modular architecture supports effective making of different products (Harlou, 2006). Efficiency is based on identifying product customisation requirements already within the product family design phase. Customisation requirements describe what kind of product variants customers are expecting. According to Pakkanen, Juuti, & Lehtonen (2016), these requirements have to be considered in defining the partitioning logic of the modular product. Partitioning logic describes the reasons why the product range is divided in a specific way into a set of modules (Pakkanen, Juuti, & Lehtonen, 2019).

The research presented in this paper is part of a larger research project that focuses on the possibilities of digitalisation in designing modular products. In this context, digitalisation refers to the development of software-based tool support for modularisation. In this paper, we focus on what to consider in the conceptual model needed in developing the tool. Section 2 considers the paper's goals in more detail. Section 3 presents the methodology used to study the topic. Section 4 presents results and Section 5 concludes the paper.

# 2 Aim

This study seeks an answer to what kind of conceptual model supports the designing modular products and the development of a dedicated tool. The objective is supported by two research questions (RQs):

RQ1. What design information elements related to modularisation can be identified?
RQ2. How do the identified design information elements relate to each other?

If it is not understood and described what kind of design information and concepts are associated with designing a modular product architecture, its documentation and management is difficult. According to our assessment, very few software-based tools have been developed specifically for modularisation. This topic is being studied to support modularisation of different products in different business environments. The overall research methodology is presented in the next section. The following section describes the steps of the entire research and how to answer the research questions.

# 3 Methodology

The aim of the overall research is to define a tool that supports modularisation and evaluate its impacts. The research is based on a constructivist research methodology. Figure 1 presents the main tasks of the research and how the tasks are connected to each other. The overall research begins with task clarification and ends with the identification of further development measures and recommendations and analysing the contribution to modularisation theories. There may be a need for iteration in the tool development based on the findings from pilot projects in which the tool will be tested. This article contributes to the research stages highlighted in red in Figure 1. Tasks highlighted in light grey are part of future research.

The objectives of the study are confirmed at the beginning of the research. This will take advantage of the literature and experiences from the previous modularisation projects. Existing tools to support modularisation are also studied at this point. Then, a description of the conceptual model for tool development is made. The description of the conceptual model begins by defining the key design information elements of the modularisation and the relationships between them. A literature review is used to identify the elements. The reference model and impact model tools of the Design Research Methodology (DRM) (Blessing & Chakrabarti, 2009) are applied in presenting the results of the literature review. The aim of reference and impact models is to describe the interplay between different elements and to show what types of elements are included in the model (e.g. findings from the literature, assumptions, experiences, or hypotheses). In this paper, all the findings of the model are based on the studied literature. According to Blessing and Chakrabarti (2009), making models like this helps to describe different design situations.

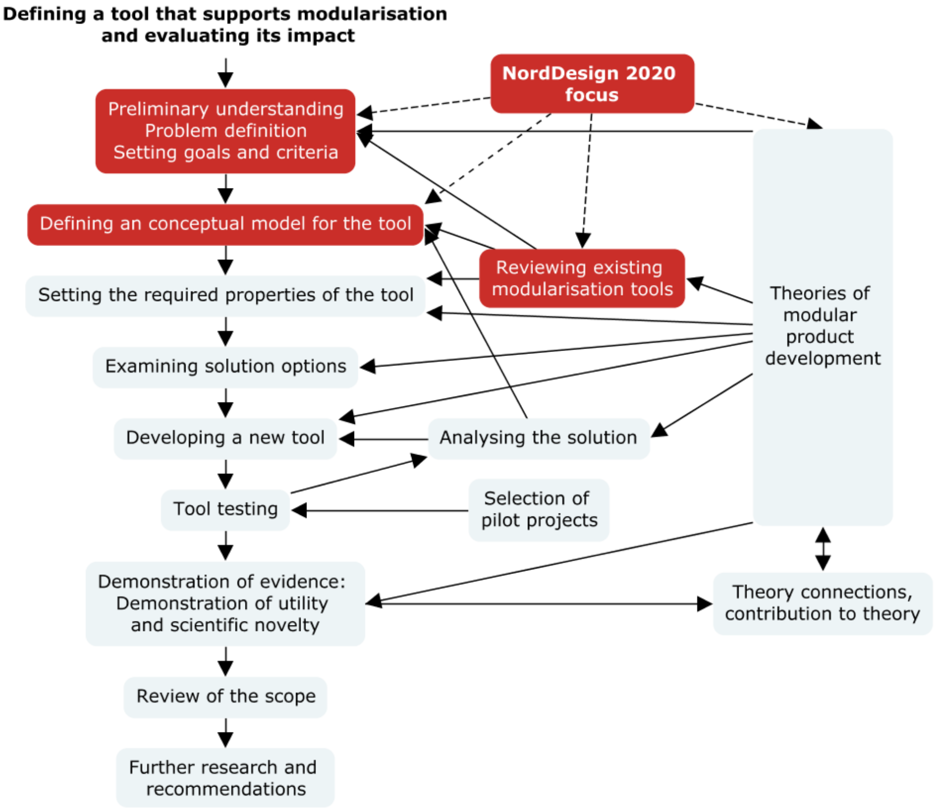

Defining a tool that supports modularisation and evaluating its impact
NordDesign 2020 focus
Preliminary understanding
Problem definition
Setting goals and criteria
Defining an conceptual model for the tool
Reviewing existing modularisation tools
Setting the required properties of the tool
Examining solution options
Developing a new tool
Analysing the solution
Tool testing
Selection of pilot projects
Theories of modular product development
Demonstration of evidence: Demonstration of utility and scientific novelty
Theory connections, contribution to theory
Review of the scope
Further research and recommendations

**Figure 1. Structure of the research and the main focus of this article.**

### 3.1 Product-related information systems in manufacturing industry

The manufacturing industry uses a number of information systems related to products. Figure 2 presents typical information systems used by the manufacturing industry and the type of information they process. Although there are many product-related information systems, there is very limited software tool support specifically for modularisation. Software commonly used by industry does not emphasise on modularisation.

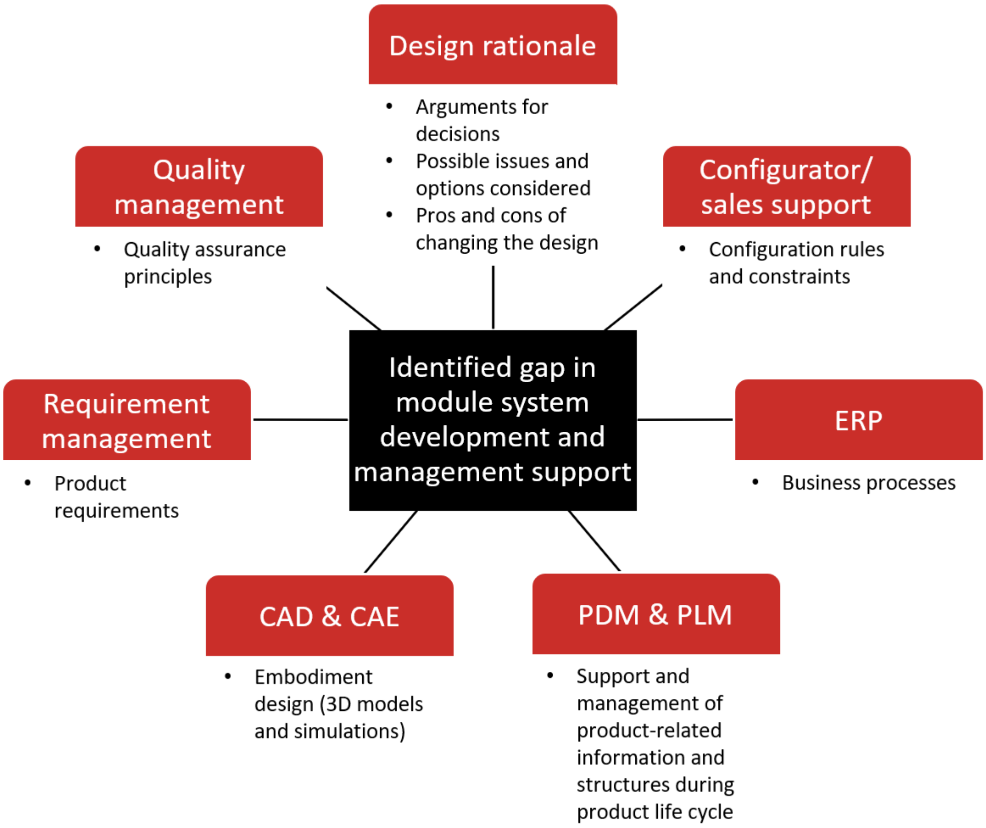

Design rationale
• Arguments for decisions
• Possible issues and options considered
• Pros and cons of changing the design
Quality management
• Quality assurance principles
Configurator/ sales support
• Configuration rules and constraints
Identified gap in module system development and management support
Requirement management
• Product requirements
ERP
• Business processes
CAD & CAE
• Embodiment design (3D models and simulations)
PDM & PLM
• Support and management of product-related information and structures during product life cycle

**Figure 2. Typical product-related information systems in manufacturing industry and the gap.**

Blessing and Chakrabarti (2009) explain, that at all stages of the design process, it is important to keep in mind the rationale behind each decision, such as possible problems, proposed solutions, and arguments for the decision. Variations of issue-based information systems (IBIS) (Kunz & Rittel, 1970) are some of the most common solutions for knowledge capturing, indexing, and retrieval.

Configurators support the user in creating product specifications by restricting how different components and properties may be combined (Haug, 2008). He classified configurators into three groups: stand-alone software shells, modules in Enterprise Resource Planning (ERP) systems, and company-specific software. ERP systems are generic, off-the shelf software packages that are aligned to the particular requirements of a company to support many key functions and business processes (Soffer, Golany, & Dori, 2003). One example of a product configurator is made by Tacton (2020).

Product Lifecycle Management (PLM) is the extension of Product Data Management (PDM) that leads toward extensive support and management of product-related information of the whole product life cycle within an enterprise (Bruun, Mortensen, Harlou, Wörösch, & Proschowsky, 2015). Commercial PLM systems include Teamcenter (Siemens, 2020c) and Windchill (PTC, 2020), for example. Computer-aided design (CAD) and computer-aided engineering (CAE) systems are tools for 3D modelling and simulation. Several commercial software tools are available, such as Solidworks (Dassault Systèmes, 2020c) and NX (Siemens, 2020a).

Requirements management tools support elicitation, analysis, specification, verification and validation, and storing of product requirements in a single dynamic repository that is accessible to project teams (Violante & Vezzetti, 2014). Commercial tools are made by, for example, Siemens (Siemens, 2020c) and Dassault Systèmes (Dassault Systèmes, 2020b). As an example, quality management software provided by Siemens (2020b) includes tools for manufacturing and ERP operations. The Siemens quality system brochures refer to product development mainly for future products, stating that the knowledge gained from manufacturing, use, and maintenance can be used in product development. Another example is the ENOVIA system (Dassault Systèmes, 2020a), which includes a section on quality management that is advertised as including, for example, support for quality auditing, management of requirements, and proactive and corrective actions.

To summarise the contribution of existing information systems to modularisation, to our knowledge, only one consulting firm has its own specific tool for modularisation (Modular Management, 2020). There has been little published research into the effects of the modularisation tool on design and management. One of the interesting research areas where tool support for modularisation is being explored is model-based systems engineering. For example, Albers et al. (2019) have developed a framework to support the modeling of modular products and systems based on SysML. They note that their framework provides a consistent way to model a modular product and supports reuse, but modeling requires time and interfaces to other systems do not yet exist.

Design rationale (DR) systems do not consider the defining and capturing of all the important issues in modularisation and do not include methodological support. However, DR aspects are important to manage the modular product in the long run. Product configurators are perhaps the closest to modularisation, but they do not guide the modularisation except by emphasising the need to study the customisation requirements. Implementing product configurator is easier when products are modularised already and the configuration rules and constraints exist (Haug, Hvam, & Mortensen, 2012). Bruun et al. (2015) suggested an approach in which the visual architecture of a product is represented and handled in a PLM system. They added that the work is still in an early phase but it is one of the few focusing on modularisation. Belkadi, Gupta, Vlachou, Bernard, & Mourtis (2016) studied the application of PLM in managing the link between the modular product structure and the selection of suppliers. Their approach is based on the fact that a number of module options are available in the PLM for all options displayed to the customers in the configurator; from these options, the designer creates the final architecture based on a set of existing and predefined architectures. Thus, the PLM research relates to modularisation, but the approaches require further development or assume that the modular system has already been developed elsewhere. CAD and CAE applications are relevant at the embodiment design stage. In terms of designing a concept of modular product, these tools go too deeply into the details and properties of solutions. The requirements management systems alone are not sufficient to capture and document the information related to modular products; more tailored support is needed. Finally, if the user wants detailed support for modularisation, the focus needs to be directed elsewhere also from the quality management systems.

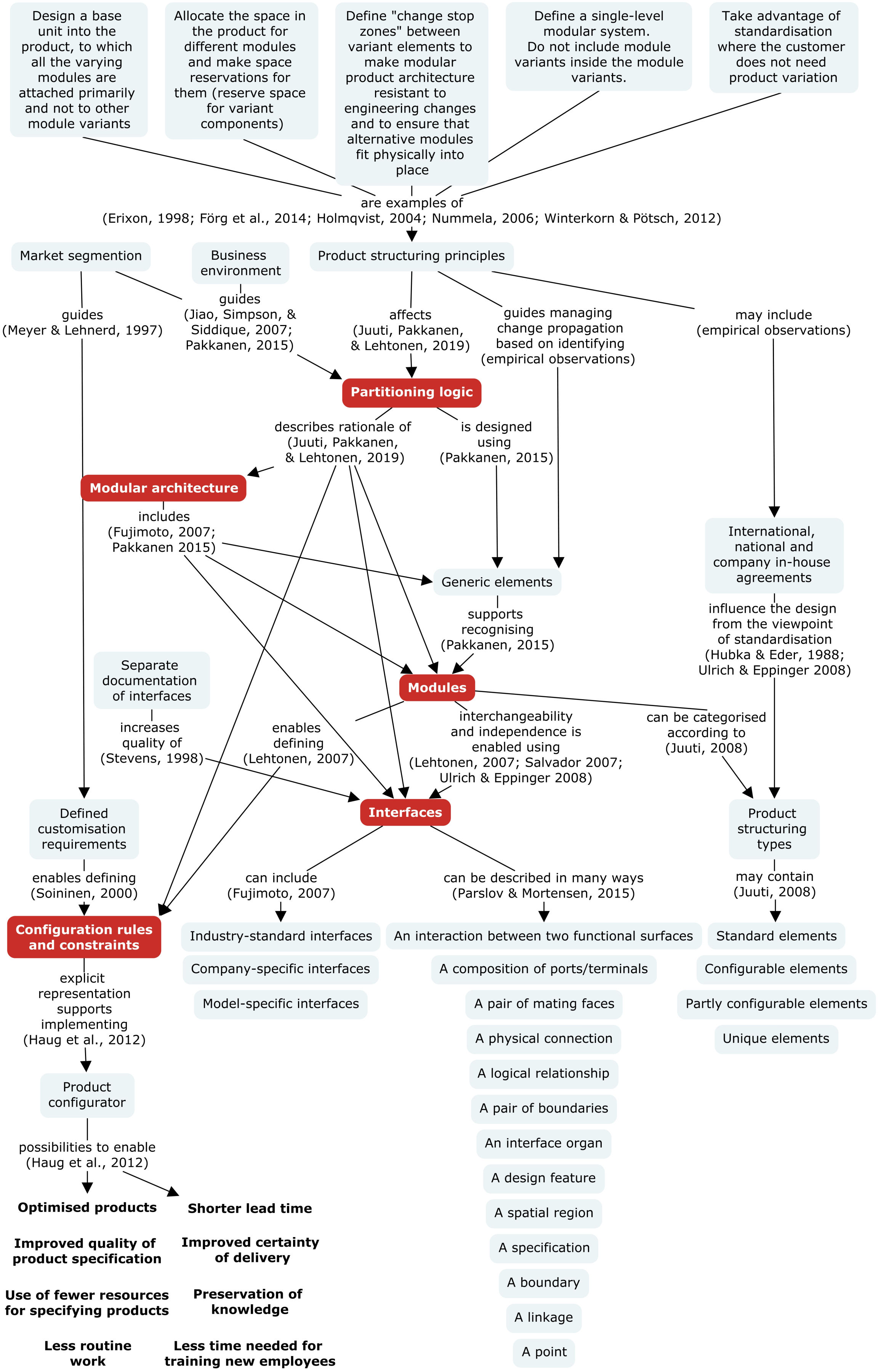
## 3.2 Design information elements of modularisation

This section deals with the key aspects in the literature that should be considered when defining the conceptual model of modularisation. Peer reviewed journal and conference articles, books and doctoral theses were identified from the selected databases (Scopus, Google Scholar, Web of Science), the most relevant publications were selected, information was collected from the publications, and finally a concise synthesis was performed applying the DRM reference and impact model presentation shown in Figure 3. Earlier publications, such as Pakkanen et al. (2019), suggested that the result of modularisation is a module system (MS) that should include descriptions of modules, interfaces, architecture, configuration rules and constraints, and, finally, partitioning logic. Figure 3 is built around these concepts (red rectangles with rounded corners), but there are other concepts that will be discussed in this section. Only selected references have been included based on the research aim. Figure 3 can be read from many different directions. The grey rectangles describe the refinements and details associated with the red rectangles. There are directed arrows between the rectangles to describe relations between elements. Relations are briefly described and references supporting each relation is presented.

Partitioning logic is one of the central concepts of modularisation. It sums up the reasons why the elements of a modular system are of a particular type (Pakkanen et al., 2019). Partitioning logic is influenced by the requirements of the business environment, including requirements for product variety from the customer's point of view (Meyer & Lehnerd, 1997; Pakkanen, 2015). For example, Jiao, Simpson, & Siddique (2007) suggest that customer viewpoints and production and supply-chain issues should be considered in development of customisable products and analysing of economic effects should be considered.

Concrete examples supporting the designing the structure of a modular system can be found in the literature. For example, Erixon (1998) talks about using basic units that always remain the same, with customisable elements attached. Another example comes from the truck industry where space reservations have been made for product architecture and layout for different module variants (Förg et al., 2014). Holmqvist (2004) proposes defining freezing zones in modular architecture to prevent the variation from spreading across the product. It has been discussed in connection with the manufacture of forest machines that a single-level module system is easier to control than if there were other variable modules inside the modules (Nummela, 2006). As a final example, cars of Volkswagen Group emphasise standardisation where no variety is needed (Winterkorn & Pötsch, 2012) We refer to these types of guidelines in this context as the product structuring principles. The purpose of these is to help find a modular product structure whose elements are capable of encapsulating product-specific customisation requirements so that the product-specific customisation needs are not spread throughout the product but remain within a particular element. These elements are referred to as generic elements.

A generic element is an supporting abstract concept for defining modules (Pakkanen et al., 2016). Juuti (2008) suggests that modules can be categorised from a configurability perspective. He identifies four main categories: standard, configurable, partly-configurable, and unique. Standard modules enable use of the same module in many products (Ulrich & Eppinger, 2008). Hubka & Eder (1988) explain that standardisation can follow from international, national, industrial or company-in-house agreements or from market trends.



Design a base unit into the product, to which all the varying modules are attached primarily and not to other module variants
Allocate the space in the product for different modules and make space reservations for them (reserve space for variant components)
Define "change stop zones" between variant elements to make modular product architecture resistant to engineering changes and to ensure that alternative modules fit physically into place
Define a single-level modular system. Do not include module variants inside the module variants.
Take advantage of standardisation where the customer does not need product variation
are examples of (Erixon, 1998; Förg et al., 2014; Holmqvist, 2004; Nummela, 2006; Winterkorn & Pötsch, 2012)
Market segmention
Business environment
Product structuring principles
guides (Meyer & Lehnerd, 1997)
guides (Jiao, Simpson, & Siddique, 2007; Pakkanen, 2015)
affects (Juuti, Pakkanen, & Lehtonen, 2019)
guides managing change propagation based on identifying (empirical observations)
may include (empirical observations)
Partitioning logic
describes rationale of (Juuti, Pakkanen, & Lehtonen, 2019)
is designed using (Pakkanen, 2015)
Modular architecture
includes (Fujimoto, 2007; Pakkanen 2015)
International, national and company in-house agreements
Generic elements
supports recognising (Pakkanen, 2015)
influence the design from the viewpoint of standardisation (Hubka & Eder, 1988; Ulrich & Eppinger 2008)
Separate documentation of interfaces
Modules
increases quality of (Stevens, 1998)
enables defining (Lehtonen, 2007)
interchangeability and independence is enabled using (Lehtonen, 2007; Salvador 2007; Ulrich & Eppinger 2008)
can be categorised according to (Juuti, 2008)
Defined customisation requirements
Interfaces
Product structuring types
enables defining (Soininen, 2000)
can include (Fujimoto, 2007)
can be described in many ways (Parslov & Mortensen, 2015)
may contain (Juuti, 2008)
Configuration rules and constraints
Industry-standard interfaces
Company-specific interfaces
Model-specific interfaces
An interaction between two functional surfaces
A composition of ports/terminals
A pair of mating faces
A physical connection
A logical relationship
A pair of boundaries
An interface organ
A design feature
A spatial region
A specification
A boundary
A linkage
A point
Standard elements
Configurable elements
Partly configurable elements
Unique elements
explicit representation supports implementing (Haug et al., 2012)
Product configurator
possibilities to enable (Haug et al., 2012)
Optimised products
Shorter lead time
Improved quality of product specification
Improved certainty of delivery
Use of fewer resources for specifying products
Preservation of knowledge
Less routine work
Less time needed for training new employees

**Figure 3. Initial reference and impact model of the key concepts in modularisation.**

Both generic elements and modules are related to modular architecture in addition to interfaces (Pakkanen et al., 2016). According to Salvador (2007), interface standardisation essentially originated in the computer industry, going back to the 1930's. Stevens (1998) emphasises making separate interface documents. With examples from the automotive industry, Fujimoto (2007) highlights the importance of understanding that interfaces can be standardised, either model-specific, company-specific, or based on industry standards. Parslov and Mortensen (2015) presents different types of definitions for an interface. These are listed in Figure 3. Lehtonen (2007) crystallises that the interchangeability and independence of the modules is ensured through the design of interfaces.

In addition to the variation requirements mentioned above, the modules are needed to define configuration rules and constraints (Lehtonen, 2007; Soininen, 2000). Haug et al. (2012) have studied the implementation strategies of configurators, and have identified several benefits if the configuration knowledge is described as a separate representation in the implementation project of the configurator. In addition, they have recognised a number of benefits associated with the use of product configurators, such as freeing of engineering resources and reduced lead time.

# 4 Results

Figure 4 illustrates a highly simplified conceptual model of the modularisation domain, generalised purely for the purposes of this article. The figure and descriptions in this section are our answers to RQ1 and RQ2. Unified Modeling Language (UML) notation is used here. UML is a commonly used way to describe concepts and the relationships between them, and the figure is all about them. The authors developed the conceptual model together with software development professionals. The concepts related to modularisation were discussed in joint workshops. At the same time, software developers documented the relations between the concepts.

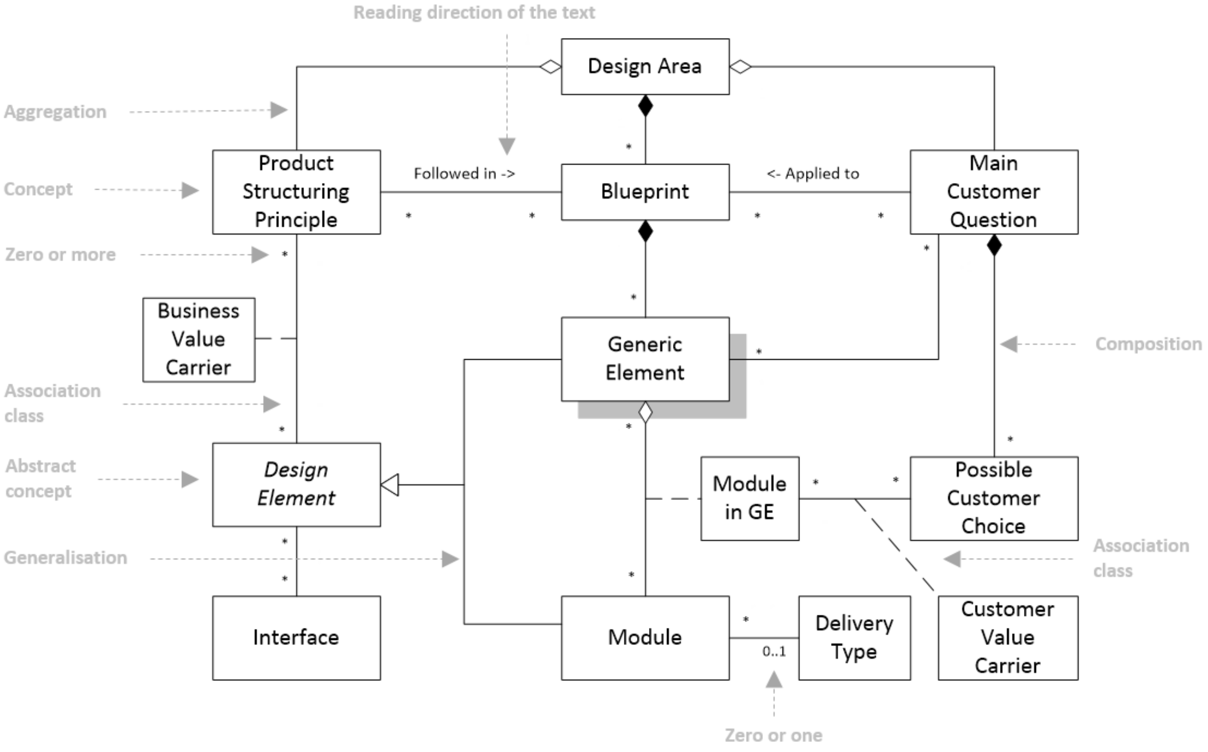


**Figure 4. Simplified conceptual model of partitioning logic in modularisation.**

The design area is a product or a smaller part of a product or, for example, a large entity such as a processing plant. An organisation may have several design areas, for example, for different product families (e.g., a camera family).

For each design area, there is a set of possible customer requirements, expressed as main customer questions. For each main customer question, there can be several possible customer choises. An example main customer question might be as follows: “What are you photographing?” The corresponding possible customer choises might be “far distance objects” and “tiny objects from a short distance.”

Product structuring principles are principles and limitations of, for example, how products are designed in the organisation. A blueprint describes a proposed partitioning logic for the design area, following the selected product structuring principles and the main customer questions.

The conceptual model has two types of design elements related to building blocks of product variants: generic elements and modules. An interface description can be attached to both design elements. Probably the most central concept in any blueprint are the generic elements. Generic elements abstract the possible varying elements in the design area based on the main customer questions and product structuring principles. The whole design can be considered in this level using generic elements, main customer questions, and product structuring principles, even when the actual module variants have not been defined yet.

Developing an module system based solely on standardised options is not always possible; instead, modular products may also need to consider delivery-specific design. Therefore, in our modeling approach, we will use the term delivery type to describe whether or not the module requires designing in the order-delivery process. Modules with different delivery types have a different impact on the lead time and cost of the order-delivery process. Delivery type is therefore a property of a module. Generic elements can consist of modules with different delivery types. The most typical delivery types are standard, configurable, partly configurable, unique, and optional. The optional module can be selected or unselected in the product.

When information about the possible customer choises is available, the actual modules can be proposed. When decisions about the design elements (generic elements and modules) are made, the design element is attached to the product structuring principles, main customer questions, and possible customer choises that led to the decision, and a reason is given for each relationship. In the conceptual model shown in Figure 4, the connection describing the reasoning between the product structuring principle and the design element is described as a *business value carrier*, whereas the relationship between the module and the possible customer choise is called the *customer value carrier*.

Conceptual model was discussed with a company producing customisable manufacturing technology. Their main criticism focused on the concept of the module, which is perhaps too simple as such and requires further structuring. The tool should support the management of module variants and their different versions. The company pointed out that if the tool is made with the current structure, user interface can become very cumbersome when all versions of the module variant are directly under the generic element. One solution to this would be to split the concept of module into generic modules and module variants. In this case, generic modules with different delivery types could be highlighted under the generic element without the structure being expanded further from the viewpoint of usability due to the different versions of module variants. Module variants would then be under generic modules in the concept hierarchy. This requires further consideration.

## 5 Conclusions

In this article, we first explored typical ICT systems used in the manufacturing industry and how those relate to modularisation. Few dedicated systems exist for the design and management of modular products. Perhaps one reason for this is that well-established modularisation practices and terminology have not emerged in the industry, although much has been done in terms of methods and publications. Such a systems could be beneficial for the conceptual design of modular products, documenting design decisions and rationale, as well as management purposes. Second, this article contributes to theories of modularisation by suggesting a conceptual model that includes concepts of design information elements considered relevant in modularisation. Preliminary discussions have been held on the rationale of the model, but from a tool development perspective, it makes sense to validate the conceptual model on a larger scale before development of more formal information models.

## References

Albers, A., Bursac, N., Scherer, H., Birk, C., Powelske, J., & Muschik, S. (2019). Model-based systems engineering in modular design. *Design Science*, *5*(May), 1–33. https://doi.org/10.1017/dsj.2019.15

Andreasen, M. M. (2011). 45 Years with design methodology. *Journal of Engineering Design*, *22*(5), 293–332. https://doi.org/10.1080/09544828.2010.538040

Belkadi, F., Gupta, R. K., Vlachou, E., Bernard, A., & Mourtis, D. (2016). Linking Modular Product Structure to Suppliers' Selection Through PLM Approach: A Frugal Innovation Perspective. In R. Harik, L. Rivest, A. Bernard, B. Eynard, & A. Bouras (Eds.), *IFIP Advances in Information and Communication Technology* (pp. 227–237). https://doi.org/10.1007/978-3-319-54660-5_21

Blessing, L. T. M., & Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. https://doi.org/10.1007/978-1-84882-587-1

Bruun, H. P. L., Mortensen, N. H., Harlou, U., Wörösch, M., & Proschowsky, M. (2015). PLM system support for modular product development. *Computers in Industry*, *67*, 97–111. https://doi.org/10.1016/J.COMPIND.2014.10.010

Dassault Systèmes. (2020a). Quality and compliance management. Retrieved from https://www.3ds.com/products-services/enovia/disciplines/quality-and-compliance/

Dassault Systèmes. (2020b). Reqtify. Retrieved from https://www.3ds.com/products-services/catia/products/reqtify/

Dassault Systèmes. (2020c). Solidworks. Retrieved from https://www.solidworks.com/

Deloitte. (2017). *Beyond the Noise: The Megatrends of Tomorrow's World.*

Erixon, G. (1998). *Modular Function Deployment - A Method for Product Modularisation*. Stockholm, Sweden: The Royal Institute of Technology.

Förg, A., Stocker, J., Kreimeyer, M., & Lienkamp, M. (2014). Enabling modularisation potentials by standardized vehicle layouts. In M. Laakso & K. Ekman (Eds.), *Proceedings of NordDesign 2014* (pp. 754–764). Espoo, Finland: The Design Society.

Fujimoto, T. (2007). *Competing to be really, really good – The behind-the-scenes drama of capability building competition in the automobile industry*. Tokyo: International House of Japan.

Hanna, M., Schwede, L., & Krause, D. (2018). Model-Based Consistency for Design for Variety and Modularization. In C. Leardi, T. R. Browning, S. D. Eppinger, & L. Becerril (Eds.), *DS 96: The 20th International DSM Conference* (pp. 239–248).

Harlou, U. (2006). *Developing product families based on architectures - Contribution to a theory of product families*. Technical University of Denmark.

Haug, A. (2008). *Representation of Industrial Knowledge - as a Basis for Developing and Maintaining Product Configurators*. Technical University of Denmark.

Haug, A., Hvam, L., & Mortensen, N. H. (2012). Definition and evaluation of product configurator development strategies. *Computers in Industry*, *63*(5), 471–481. https://doi.org/0.1016/j.compind.2012.02.001

Holmqvist, T. (2004). *Managing Product Variety through Product Architecture*. Chalmers University of Technology.

Hubka, V., & Eder, W. E. (1988). *Theory of technical systems: a total concept theory for engineering design*. Berlin: Springer-Verlag.

Jiao, J., Simpson, T. W., & Siddique, Z. (2007). Product family design and platform-based product development: A state-of-the-art review. *Journal of Intelligent Manufacturing*, *18*(1), 5–29. https://doi.org/https://doi.org/10.1007/s10845-007-0003-22

Juuti, T. (2008). *Design Management of Products with Variability and Commonality - Contribution to the Design Science by elaborating the fit needed between Product Structure, Design Process, Design Goals, and Design Organisation for Improved R&D Efficiency*. Tampere University of Technology.

Kunz, W., & Rittel, H. W. J. (1970). *Issues as elements of information systems*.

Lehtonen, T. (2007). *Designing Modular Product Architecture in the New Product Development*. Tampere University of Technology, Tampere.

Meyer, M. H., & Lehnerd, A. P. (1997). *The Power of Product Platforms: building value and cost leadershop*. New York: The Free Press.

Modular Management. (2020). PALMA. Retrieved from http://palma-software.com/

Nummela, J. (2006, February 24). *Integrated Configuration Knowledge Management by Configuration Matrices – A Framework for Representing Configuration Knowledge* (p. 205). p. 205. Retrieved from http://dspace.cc.tut.fi/dpub/handle/123456789/209

Pakkanen, J. (2015). *Brownfield Process: a method for the rationalisation of existing product variety towards a modular product family*. Tampere University of Technology.

Pakkanen, J., Juuti, T., & Lehtonen, T. (2016). Brownfield Process: A method for modular product family development aiming for product configuration. *Design Studies*, *45*, 210–241. https://doi.org/10.1016/j.destud.2016.04.004

Pakkanen, J., Juuti, T., & Lehtonen, T. (2019). Identifying and addressing challenges in the engineering design of modular systems – case studies in the manufacturing industry. *Journal of Engineering Design*, *30*(1), 32–61. https://doi.org/10.1080/09544828.2018.1552779

Parslov, J. F., & Mortensen, N. H. (2015). Interface definitions in literature: A reality check. *Concurrent Engineering*, *23*(3), 183–198. https://doi.org/10.1177/1063293X15580136

PTC. (2020). Windchill. Retrieved from https://www.ptc.com/en/products/plm/plm-products/windchill

Salvador, F. (2007). Toward a Product System Modularity Construct: Literature Review and Reconceptualization. *IEEE Transactions on Engineering Management*, *54*(2), 219–240. https://doi.org/10.1109/TEM.2007.893996

Siemens. (2020a). NX. Retrieved from https://www.plm.automation.siemens.com/global/en/products/nx/

Siemens. (2020b). Quality Management Systems (QMS). Retrieved from https://www.plm.automation.siemens.com/global/en/our-story/glossary/quality-management-systems-qms/38124

Siemens. (2020c). Teamcenter. Retrieved from https://www.plm.automation.siemens.com/global/en/products/teamcenter/

Soffer, P., Golany, B., & Dori, D. (2003). ERP modeling: a comprehensive approach. *Information Systems*, *28*(6), 673–690. https://doi.org/10.1016/S0306-4379(02)00078-9

Soininen, T. (2000). *An Approach to Knowledge Representation and Reasoning for Product Configuration Tasks*. Helsinki University of Technology, Espoo.

Stevens, R. (1998). *Systems Engineering: Coping with Complexity*. Retrieved from https://books.google.com/books?hl=fi&lr=&id=PPBp2RwMFWwC&pgis=1

Tacton. (2020). Design Automation. Retrieved from https://www.tacton.com/design-automation/?via=header

Ulrich, K. T., & Eppinger, S. D. (2008). *Product Design and Development*. New York: McGraw-Hill.

Violante, M. G., & Vezzetti, E. (2014). A methodology for supporting requirement management tools (RMt) design in the PLM scenario: An user-based strategy. *Computers in Industry*, *65*(7), 1065–1075. https://doi.org/10.1016/J.COMPIND.2014.05.001

Winterkorn, M., & Pötsch, H. D. (2012). *Volkswagen Golf VII: Launch of a new era*. Sardinia.