



AALBORG UNIVERSITY
DENMARK

Aalborg Universitet

Product Family Modeling

Working with Multiple Abstraction Levels

Jørgensen, Kaj Asbjørn

Published in:

Handbook of Research in Mass Customization and Personalization

Publication date:
2009

Document Version
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Jørgensen, K. A. (2009). Product Family Modeling: Working with Multiple Abstraction Levels. In F. T. Piller, & M. M. Tseng (Eds.), *Handbook of Research in Mass Customization and Personalization* (Vol. 1, pp. 319-337). World Scientific.

General rights

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

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

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Product Family Modelling

Working with multiple abstraction levels

Kaj A. Jørgensen

Aalborg University, Dept. of Production, Denmark

kaj@production.aau.dk

Application of product configuration in manufacturing-to-order (MTO) companies and engineer-to-order (ETO) companies is significantly different compared to mass-producing companies. Furthermore, the situation is often made extra difficult by market conditions, which imply long order horizons and many changes of orders both before and after order acceptance.

With focus on these challenges, a special approach is presented for modelling of product families on multiple abstraction levels. With this approach, customer driven product configuration is concentrated on decisions, which are relatively invariant throughout order processing. Higher abstraction levels are typically related to identification of basic functionalities of the product and considerations about the ability to perform functions, which are required by the customer. They are very primary and should clearly be addressed in sales and tendering. By the proposed modelling approach, it is shown how the focus of product configuration can be shifted to identification and definition of attributes instead of modules and components. It is also shown that classification is a means for identification of multiple abstraction levels.

Keywords

Mass customisation, product configuration, product model, product family model, engineer-to-order, abstraction level, classification, composition, object-oriented analysis and design.

1. Introduction

Product Configuration and Product Family Modelling have been important topics since Mass Customisation (MC) was initiated more than one decade ago. This research topic was initiated with Davis' publication "From Future Perfect: Mass Customisation" (Davis, 1989) and it has been proved, how products and services can be realised as a one-of-a-kind manufacture on a large scale. Davis also presented the idea that the customisation could be done at various points in the supply chain. Later, in 1993, Pine published a major contribution to the mass customisation literature: "Mass Customization: The new Frontier in Business Competition" (Pine, 1993), (Pine et al., 1993), which was an extensive study of how American enterprises during the seventies and eighties had been overrun by the efficient Japanese manufacturers, which could produce at lower costs and higher quality. Since its introduction, MC has called for a change of paradigm in manufacturing and several companies have recognised the need for mass customisation. Much effort has been put into identifying, which success factors are critical for an MC implementation and how different types of companies may benefit from it (Lampel and Mintzberg, 1996), (Gilmore and Pine, 1997), (Sabin, 1998), (Silveira et al., 2001), (Berman, 2002), (Silveira et al., 2001).

2. Product Family Modelling – Various Approaches

The fact that products must be easily customisable in order to achieve MC has been described comprehensively in the literature. In (Berman, 2002) and (Pine, 1993), it is argued that the use of modular product design combined with postponement of product differentiation would be an enabler to a successful MC implementation. This issue of course also relates to readiness of the value chain.

Traditionally, a product family can be viewed as the set of end products, which can be formed by combining a predefined set of modules (Faltings, 1998), (Jørgensen, 2003). This set of products is considered as a whole and forms a product family. The product family is modelled as one single model and describes which modules are parts of the product family model and how they can be combined. When a product family model is implemented in a configurator, users are allowed to select modules to configure products, and in some cases the user can even select the desired properties of the end product and the configurator selects the corresponding modules (Jørgensen, 2003). Several different methods for defining product models have been constructed during the latest years, each with their own advantages.

In (Hvam, 1999) and (Hvam, 1994), a "Procedure for building product models" is described as a very practical approach with a seven step procedure, describing how to build a configuration system from process and product analysis to implementation and maintenance. For the product modelling purpose, the Product Variant Master method is used to produce an overview the generic product structures and possible variants. This is followed by object-oriented modelling to describe both classification and composition in a product family. The object-oriented approach is also applied by (Felfernig et al., 2001), who uses the Unified Modelling Language (UML) to describe a product family. This is done by using a UML meta

model architecture, which can be automatically translated into an executable logical architecture. In contrast to (Hvam, 1999) this method focuses more on formulating the object-oriented product structure, rules and constraints most efficiently. The method also focuses on how the customers' functional requirements can be translated into a selection of specific modules in the product family.

Mapping of functional requirements to specific modules is considered in (Jiao et al., 1998) and (Du et al., 2000), where it is proposed to use a triple-view representation scheme to describe a product family. The three views are the functional, the technical and structural view. The functional view is used to describe, typically the customers, functional requirements and the technical view is used to describe the design parameters in the physical domain. The structural view is used for performing the mapping between the functional and technical view as well as describing the rules of how a product may be configured. The description of this modelling approach is however rather conceptual, and it is not easily implemented in common configuration tools.

Most of the methods, which exist for modelling configurable products, focus on modelling the solution space of a configuration process. This means that they do typically not focus on information which is not directly used to perform the configuration itself. This information could include e.g. customer, logistics and manufacturing information according to (Reichwald et al., 2000). Here also, the emphasis is put on the importance of managing these flows efficiently, which is most likely to be done by building an integrated information flow. In order to do this, the information must be structured in an appropriate way, which can be done by constructing an information model.

There are different strategies on how to construct the most appropriate information models, and they naturally also varies between different companies, markets and products. But even though there is not a single generic strategy from which the optimal information model can be constructed, the importance of this issue must be emphasised. Since most of the methods, which are developed for MC and product modelling, have been developed for mass producing companies, these methods are not always easily applicable to other production set-ups. In the following section, some of the difficulties associated with doing this in engineer-to-order companies will be introduced by a description of a case company. In this case description, problems regarding the implementation of MC and product configuration will also be described and related to the field of information modelling.

3. Product Modelling in an ETO Company

Implementation of Mass Customisation and product configuration in engineer-to-order companies is significantly different compared to mass producing companies.

Aalborg Industries is the world leading manufacturer of steam boilers for marine applications. The headquarter is located in Denmark, but sales offices are located all over the world, and there are production facilities in Denmark, Brazil, Indonesia, China and Vietnam. Aalborg Industries has around 1650 employees worldwide, and had in 2005 a turnover of around 200 million €. It is a typical engineer-to-order

company, where each order is engineered to meet specific customer requirements. During the recent years Aalborg Industries has been modularising the products and developed a configuration system, which is implemented and working today. In the following, a few issues, which illustrate the central problems regarding the management of information in a company like Aalborg Industries, have been selected. The focus will be put on problems regarding information flow i.e. registration, structuring and usage of information.

Product development in the global perspective is carried out at the headquarter in Aalborg and the specifications for the new products are distributed to the sales offices and production facilities worldwide. Regarding management of information, this presents a challenge in adapting new products to the local standards in relation to available materials and manufacturing capabilities. Also when implementing the configuration system for a new product, the variance must be defined in such a way that it can satisfy the needs of customers at all locations in the world. Information is often transferred between the different companies, when e.g. complex engineering work must be done on a boiler, information is transferred from a local sales office to Aalborg, where the majority of engineering knowledge is located. Here the engineering work is done, and information is transferred back to the sales office and subsequently to the manufacturing department. Regarding different languages and standards, this presents a number of problems if functionality supporting this is to be implemented in a configuration system. There is a very large variance between the products and for each order this results in a very large amount of information to be handled.

Through the order processing from initial contact with the customer until the boiler products are delivered, the order data passes through a great number of departments. If a configuration system is to support the business processes fully from sale to delivery, this sets new requirements to the way information is handled and presented, since different information is needed in the different tasks during the processes. This is further complicated due to the requirement, that during these processes some information, which other processes depend upon, may be changed. Examples of such external changes are changes in prices for raw materials and transportation, subcontractor and supplier availability and currency exchange rates. Changes in any of these factors may create an incentive to change the configuration itself or other product information. If information describing the product is changed, then it is important that the changes are reflected in the information presented in all other processes which depend on this information.

The time from an initial request from a customer to the delivery of a boiler plant spans a long time, some times even years and a configuration may be changed a number of times during the sales process, as well as by the customer even after the sales contract has been signed. This provides further challenges to the above mentioned problem on changes in information.

Another central issue in the above mentioned problem areas, is the mapping between requirements and components. As an example, a number of pumps are configured in the configuration system to be a part of a product but, depending on where the product is to be manufactured and delivered. There are different reasons why it may be more optimal to select different brands or suppliers of the pumps. This is also a great challenge in handling information, since information regarding

the specifications of the pumps must be combined with information about local suppliers, to determine if a pump should be bought locally or elsewhere. If the information could be structured in a way that allowed a mapping between specifications and local availability; it would also provide an opportunity to optimise the configurations with respect to prices or other criteria.

The description of this case company reveals some of the challenges that Aalborg Industries will face in the future regarding product configuration. It also indicates clearly that there is a big difference between how mass producing companies can take mass customisation into account compared to engineer-to-order companies. Overall, there is a need to concentrate on specification of relatively invariant requirements in the sales process and postpone e.g. the selection of specific components and suppliers as long as possible. This would give the freedom to select the most appropriate components regarding e.g. price as well as make it easier to handle changes late in the process.

4. Product Family Models

It is characteristic for a product family model that it has a set of *open specifications*, which have to be decided to determine in order to configure an individual product in the family (Jørgensen, 2003). The product family model serves as a foundation for the configuration process (see figure 1) and, in order to secure that only legal configurations are selected, the family model should contain restrictions about what is feasible and what is not. Hence, the product family is the set of possible products, which satisfy the specifications of the product family model. The result of each configuration is a *model of the configured product*. From this model, the physical product can be produced (see figure 1). So, ideally, each product model must have sufficient data about attributes and structure in order to manufacture the physical product. A *product configurator* is defined here as a tool, computer software, which is built on the basis of a product family model and which can support users in the configuration process (Faltings, 1998).

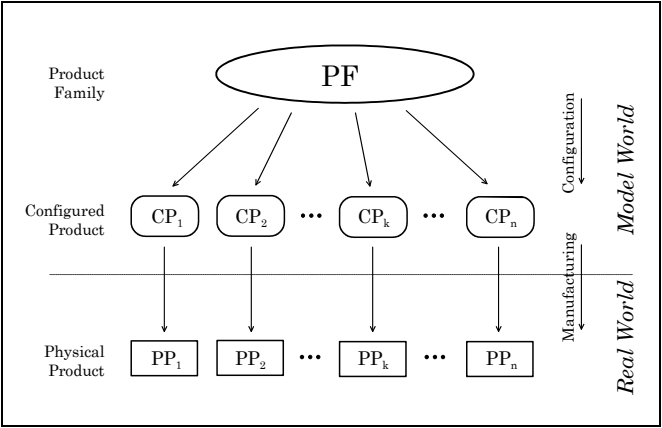


Figure 1 - The product family model as the foundation for product configuration.

Product configuration in the simplest form is a matter of combining a set of *modules* so that the product model contains information about what modules and components are to be assembled. In this *compositional view*, a product consists of a number of components, which subsequently can consist of other components, etc. Modules are identified on a level above components from a configuration point of view whereas components usually are identified from a manufacturing point of view. Most often, the number of modules is smaller than the number of related components. Thus, in the *structural model* for configurable products, products consist of modules and modules consist of other modules and/or components.

In connection with identification of modules, it is important to analyse how modules interface with each other. Therefore, it is important also to look at the modules functional characteristics and secure that the modular structure is harmonised with the functional division of the product (Andreasen, 2003).

Besides structure, products have *properties*. It is essential for both the customer and the producer to focus on properties of the resulting product. For each configured product, the resulting properties are dependent of the selected components and structure of the product. In the product configuration process, algorithms must be available to estimate the resulting product properties. Some properties are simply the properties of the components, e.g. the colour of a car is normally defined as the colour of the car body. Other properties are computed from properties of the components. For example, the weight is simply the sum of the component's weight. However, not all resulting properties are so easy to determine and rather complicated relationships exist. For instance, the resulting performance of a pump is a non-linear function of certain component properties.

In the following, the term *attribute* will be used in the models corresponding to properties of physical products. Consequently, when a configuration is performed, the desired properties of the resulting product must be determined by defining values of attributes in the product family model. All relevant attributes of both the resulting product and the available modules must be specified and their optional values to be selected during configuration tasks must also be defined. In (Jørgensen, 2008), the content of product family models is described in further detail and examples are shown by use of a simple synthetic language.

5. Information Modelling - A Generic Model Component

Methodologies for system development are often based on concepts derived from General Systems Theory (Skyttner 2005). According to this theory, a *system model* is an intentionally simplified description of a system, fulfilling a certain purpose. Hence, the simplifications imply that some choices are made in order to select the most important properties, components and relationships. Thus, a system model can e.g. be suitable for communication between designers, because with the model, it will be possible to concentrate on the most important aspects of the system. Models are viewed either as *analysis models* or *synthesis models*. Analysis models are models of something existing, often physical objects and synthesis models are models created as a foundation for construction of something new, which eventually will become physical – an artefact (Jørgensen, 2002). Hence, synthesis

models are built from ideas, thoughts and imaginations and obtained in some kind of representation. Design by modelling is a development approach, where a synthesis model is designed as an intermediate result and the final result is an implementation of the model in the real world.

In order to be able to create all sorts of models and to perform many different modelling processes, a conception of a *generic model component* has been introduced (Jørgensen 2005). This component is inspired from general systems theory and from object-oriented modelling and can be regarded as a component that can be used for system models in general and for information modelling.

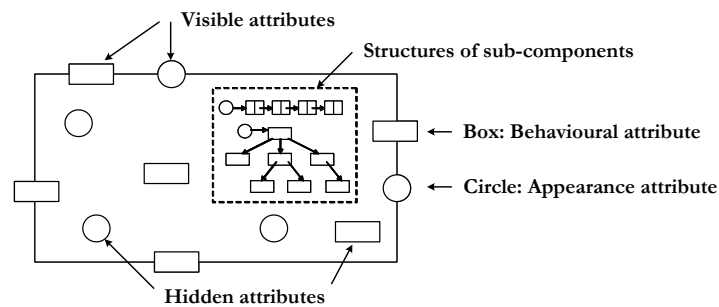


Figure 2 - Generic model component

The generic component consists of a set of *attributes* and a *structure of sub-components* (see figure 2). Some attributes are *factual attributes*, defining the state of the component, and some attributes are *behavioural attributes*, defining the operations, which the component can carry out. An alternative division of attributes defines some attributes as *visible attributes*, which can be called from other components, and some are defined as *hidden attributes*. The structure establishes the relationships between the component itself and the sub-components. All sub-components are regarded the same way, recursively. With this generic component, it is possible to address the following important issues of top-down system modelling: purpose, function (visible behavioural attributes), form (visible factual attributes), internal (hidden) attributes and internal structure.

All structures can be represented by two kinds of relationships in the information model (Jørgensen, 1998): *references* (one-to-one relationships) and *collections*, (one-to-many relationships). For e.g. a computer, a reference could represent the relationship e.g. between the keyboard and the computer. A collection could represent the relationship e.g. between the cpu board, the anchor, and multiple memory units, the members.

When a synthesis information model is considered, a foundation for the components must be established by creating *types* of components. Component types are the primary content of information models and components are generated from component types (see figure 3). It is important to distinguish between modelling on the *object level* and modelling on the *type level*.

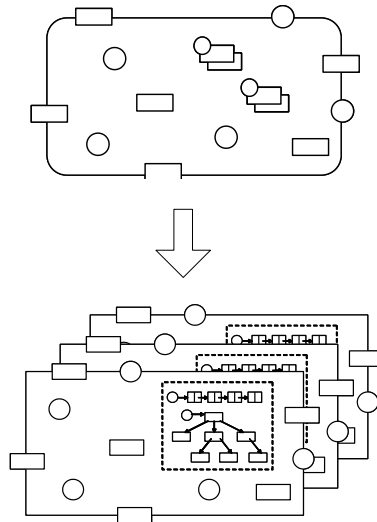


Figure 3 – Component type is the basis for generating components (instances)

An important fundamental issue of information modelling is *abstraction mechanisms*, which provide the means for identification and design of invariant components and structures (Smith 1977a), (Smith 1977b), (Rosch 1978) and (Sowa 1984). Two abstraction mechanisms are defined here: *composition* and *classification* (Jørgensen, 1998). Composition focuses on the components and the relationships between the components. The most frequently used structure is the component structure, which shows *aggregation* versus *separation*. Such a structure is illustrated in figure 4 for a sample computer.

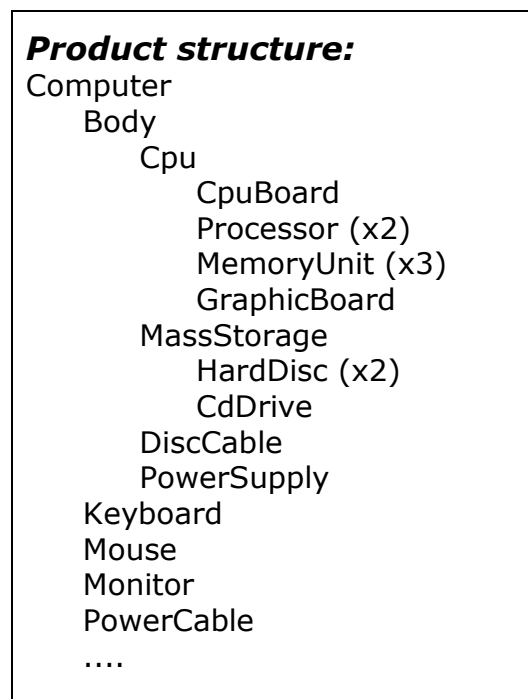


Figure 4 – Sample composition structure of a computer

Classification, on the other hand, focuses on identification of *classes/types* of components based on the *properties/attributes*, which characterise them. This can

be illustrated in a diagram, which is termed a *taxonomy* (see figure 5), where the relationships *generalisation* versus *specialisation* are shown. Often, a UML class diagram is used for the taxonomy ((Rumbaugh et al. 1999)).

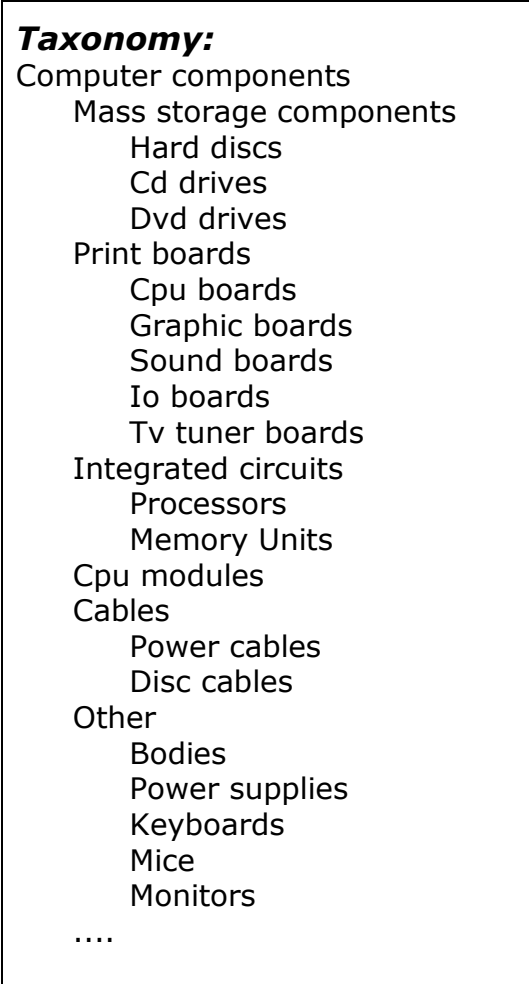


Figure 5 - Sample taxonomy of computer components

In information modelling, composition and classification together support identification of fundamental structures on a type level as the basis for generation of individual components on an object level and they provide the means to set particular focus on the most invariant decisions. A classification process results in a basic structure of types and a composition process results in a basic structure of components.

When both abstraction mechanisms are used in design tasks, then, as indicated in figure 6, classification is used first and composition afterwards. Classification primarily supports the identification of model components and the basic structure at the type level. Based on this, the structural considerations are identified by use of composition.

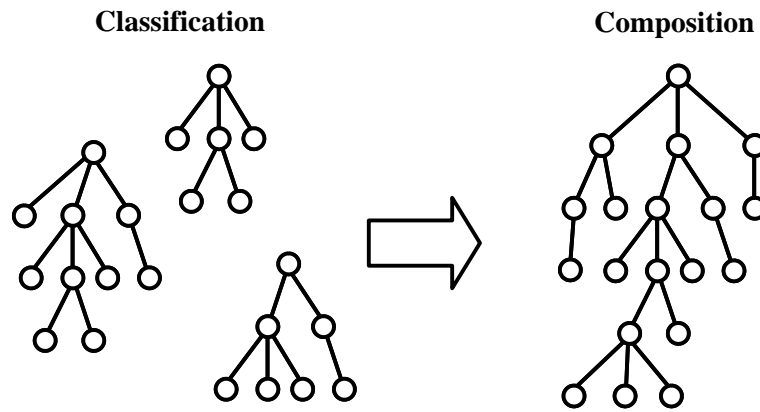


Figure 6 – Classification and composition hierarchies

Each component type includes a specification of a set of attributes with *name* and *data type*. The classification abstraction mechanism is primary because, based on attributes, the component types can be classified and organised in a hierarchy, the taxonomy. Identification and specification of structures can also be included in the component types by creating the *relations*, which formulate the *constraints* regarding attributes and combinations of sub-components (see figure 7). The component type is a kind of template and, from each type, an indefinite number of components, instances, can be generated. The quality of these component types is the key basis to achieve an invariant information model foundation.

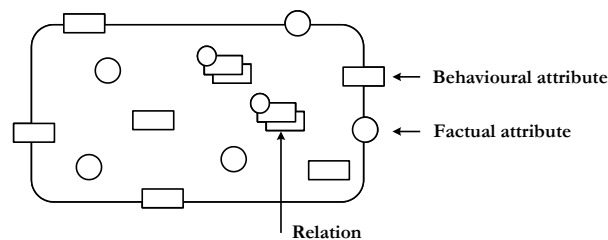


Figure 7 – Component type with relations/constraints

6. Product Family Models – Attributes and Modules

The basic units of a product family model are module types (Jørgensen, 2008). A *module type* is a model of a set of modules, which are interchangeable, perhaps with some restrictions. With reference to the previously mentioned compositional view, individual modules of each type are selected, when configuration is performed. The attributes of the module types are selected on the basis of what is important and relevant for the end-product. In fact, modules can be determined from attributes.

When products are installed in their user environment, they perform their functions – hopefully in the expected way. Therefore, considerations about the ability to perform the functions, which are required by the customer, are very important and

should be a significant subject of configuration. Hence, the focus of product configuration is shifted to identification and definition of product attributes instead of modules and components. This is particularly important in companies, where order horizons are long and where many changes often have to be managed.

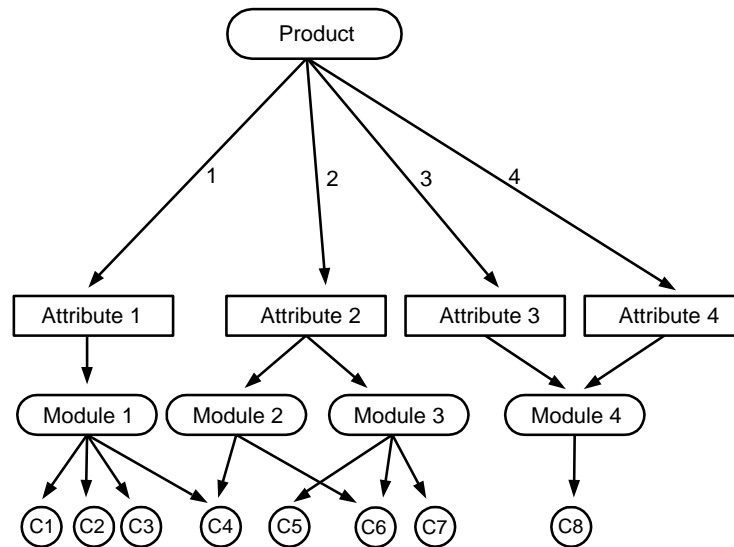


Figure 8 Specification of modules indirectly through attributes. Attribute 1 corresponds to one module whereas attribute 2 determines two modules. In contrast, module 4 is determined by two attributes.

Figure 8 illustrates how underlying modules/components of an end-product in a product family can be determined on the basis of decisions regarding attributes. Attribute 1 corresponds to one module whereas attribute 2 determines two modules. Further, the figure shows that module 4 is determined by two attributes.

If this idea is applied to the computer example, all choices about internal modules of the computer must be transformed to attributes. For instance, instead of selecting hard disks directly as sub-modules, a set of attributes must be identified and defined to provide the same possibilities. An attribute 'DiskMemory' could represent the total storage capacity of the contained disks and a logical attribute 'MinimizeDiskPrice' could be used to indicate that the price should be minimised. Furthermore, attributes about quality ranking could be added. As a result, the most suitable disk or disks could then be selected automatically based on the values of the attributes.

With this in mind, it can be stated that the configuration process can be considered as a mixture of attribute specification and selection of modules, which together must satisfy the required attribute values. Consequently, the internal structure can be hidden and decisions about the internal structure can be postponed. Thereby, higher levels of abstraction can be identified by focusing on attributes instead of structure.

7. Product Family Models – Abstraction by Classification

Regardless of whether the selection of modules is implicit or explicit, multiple abstraction levels can also be established by the use of classification. In a taxonomy of module types (see figure 5), the types towards the root are the most general types whereas the types towards the leaves are the most special types. Therefore, a selection of relatively general types represents a higher abstraction level compared to selection of relatively special types.

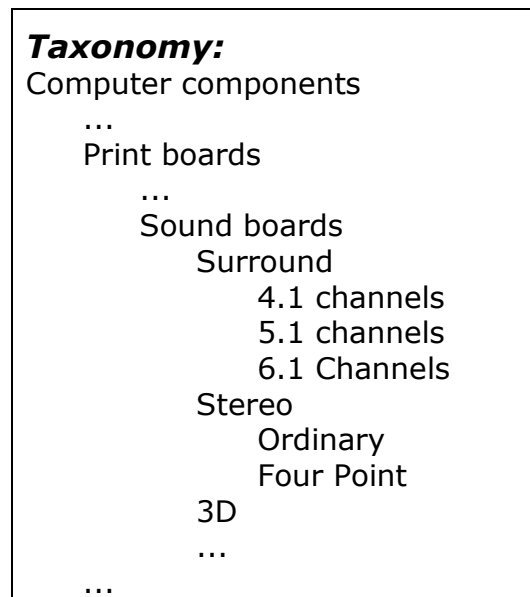


Figure 9 – Further classification of sound boards

Figure 9 shows a partial taxonomy as a further classification of a specific module type of figure 5 and reveals two additional levels of specialisation. Clearly, this example illustrates that a preliminary selection of a relatively general type is a way of postponement, i.e. some indications are given but further specifications can be submitted.

All module types have attributes, which can be included in the configuration process. Besides an obvious price attribute, further technical properties of the available modules can be represented as attributes of the module types. These attributes can be located at different levels of the taxonomy depending on how general or special they are. Consequently, a selection of a type results in a set of additional attributes, which can be used for further specification. However, if a specification of a specific attribute is required, a specialisation down to a certain level is implicitly made. If for instance something is required about attributes which are only relevant for stereo sound, then stereo sound boards are implicitly selected.

In general, classification is highly related to attributes. Besides what is already described, identification of sub-modules can be based on values of attributes. For instance, the sub-types of surround sound board could be identified by values of an attribute 'NoOfChannels'. In fact, this attribute could remove the need for classification at the lowest level. Hence, if multiple classifications of these sound boards were relevant, i.e. if multiple and equally important classification criteria

exit, it will be more flexible to identify the corresponding attributes and their possible values.

8. Application of Product Family Modelling

Many observations indicate that implementation of Mass Customisation and product configuration in ETO companies must focus on product modelling in order to gain immediate economic results from saving resources for tendering and order processing. This top-down development approach is also important when different organisational units must be joined and different software applications and databases must be integrated. Therefore, a number of theoretical topics about system modelling, product modelling, modelling of product families, information modelling and data modelling must be utilised.

In this chapter, it is proposed that modelling of product families should be performed in a way that multiple levels of abstraction can be identified and a top-down configuration approach with specification of attributes and structure. This is especially suitable for order processing over long time, where it is important to control the degree of freedom at different steps. It is necessary to postpone certain decision until enough requirements are available.

The proposed approach is currently under implementation at the Danish case company, Aalborg Industries. Here, the development of product family models and product configurators has been carried on for several years starting with a simple model for calculation of quotations. In later versions, data from the product configurator has been used as parameter input to other software applications for producing data sheets and drawings. This development has proved the necessity to set greater focus on product modelling on multiple abstraction levels.

The current version of the product configurator is web-based so that sales and tendering can take place everywhere around the world. This technology will also be used in the future and the company is now developing a more advanced product model and related product configurator software modules with the purpose of integrating more of the existing software applications and get more optimised order processing and production planning. Furthermore, supply chain management issues are taken into consideration so that decisions about selection of manufacturing locations and suppliers can be optimised. Especially, issues about interaction with ERP systems are important and require software modules for automatic interfacing.

As described for the case company, the order horizon can be rather long and many changes in the order specification occur. In addition, many modules can be purchased as products from multiple suppliers, which can deliver a variety of properties for sizes, price, performance, quality, lead time, etc. Hence, for this company, it will be important to rise to a higher abstraction level by setting focus on specification of attributes and move away from the structural model of configuration.

Two examples from the case company can illustrate this. In the first example, alternative feed water pumps for boilers can be selected as illustrated in table 1.

Table 1 – Alternative feed water pumps specified with a set of attributes

	Delivery head Bar(gauge)	Capacity m ³ /h	Supply voltage V	Price €
(Requirements)	(≥ 22)	(≥ 25)	(3 x 330)	
Product 1	23	25.5	3 x 330	1600
Product 2 *)	25	30	3 x 330	2000
Product 3 **)	24	25	3 x 330	1800

*) Has frequency converter drive, i.e. significantly lower power consumption

***) Approved for running in explosion risky zones

Table 1 shows that three sample requirements are specified and that three different pump products can satisfy the requirements. It also shows that additional attributes may be taken into consideration if further specifications have to be made.

In the second example (see table 2), it is shown that alternative safety valves can be selected.

Table 2 – Alternative feed water pumps specified with a set of attributes

	Set pressure Bar(gauge)	Size	Production location	Delivery time	Price €
(Requirements)	(19)	(DN50)	(Deliv. location: Finland)		
Product 1	19	DN50	Germany	2 days	200
Product 2	19	DN50	China	30 days	130

Two valve products satisfy the requirements but, as shown, with great difference between the prices. A significant attribute is the delivery time, which may set serious limitations regarding the time for procurement. However, this is dependent on the production location so, if for instance the production location is changed to the East Asia, a dramatic reduction of delivery time and price can be reached.

Two examples of abstraction by classification can also be presented (see (Jørgensen, 2008) for description of the syntax). Example one is about oil fired boilers, where the module type 'OilfiredBoiler' is the super-type for two sub-types 'MissionOS' and 'MissionOL'. Two attributes show the decision making, 'BurnerType' and 'Capacity'.

```

type OilfiredBoiler
{
  BurnerType : {KB,KBO,KBE,KBSA,KBSD};
  Capacity : (1.6 .. 15.5);
}

```

```

type MissionOS subtypeof OilfiredBoiler
{
  BurnerType : {KB,KBO} default KB;
  Capacity : (1.6 .. 6.5);
  ...
}

```

```

type MissionOL subtypeof OilfiredBoiler {...}
Etc.

```

For oil fired boilers, the burner type can be any of the listed values, while for mission OS boilers only a subset of burners is valid. The capacity for mission OS boilers is similarly narrowed compared to the oil fired boilers in total.

Example two regards feed water pump units, where there are two sub-types and where the regulation type differs.

```

type FeedWaterPumpUnit
{ RegulationType : {OnOff,Modulating}; }

```

```

type FeedWaterPumpUnitOnOff subtypeof FeedWaterPumpUnit
{ RegulationType : {OnOff}; }

```

```

type FeedWaterPumpUnitModulating subtypeof FeedWaterPumpUnit
{ RegulationType : {Modulating}; }

```

Both examples show that the super-type modules represent decisions on a higher abstraction level because selection of a general module type establish some degree of specification while remaining decisions are postponed. In contrast, sub-types represent decisions about more precise specifications. In the sales process, it will be possible to assist the customers with decisions about how specific they must be from the beginning. A balance must be obtained. Relatively specific decisions give more precise estimations (cost, required capacity, delivery, etc.) but are most likely subject to changes and, on the other hand, decisions on a more general level will lead to uncertainty about estimations. A key issue in relationship with configuration is to develop models for calculating estimations based on different levels of abstraction in decision making.

9. Conclusion

In this chapter, it is underlined that there are some fundamental issues of information modelling, which can be applied to product family modelling. For Product family models, it is important to identify the attributes in the model of the

end-products and, because some attributes in models of product families will be assigned values during the configuration process, they must be defined with optional values i.e. domains. It is also characteristic for product family models that relations/constraints must be defined between attributes of the possible end-products and the attributes of the identified modules/components.

In the chapter, there is set special focus on how to develop product family models, which can support product configuration on multiple abstraction levels. First of all, it is proposed that configuration is performed by specification of attributes instead of selection of modules. This means that the structure of end-products is defined indirectly based on the values of attributes. Thereby, configuration is also oriented towards customer needs because attributes are essential in connection with the functional demands from customers. Further, it is proposed that, when modules are selected, it is important to develop classifications of module types and form a taxonomy. Such a structure is well suitable for identification of multiple abstraction levels by classification, where specifications can range from a general level to a more specific level.

The aim of developing product family models is that they can be used as a foundation for development of specific product configurator software and the proposed methodology, included in this chapter, is for the moment being used by a particular ETO company, which intend to develop an advanced product family model and a product configurator that can support many organisational functions in the company world wide. Especially, the top-down approach with modelling on multiple abstraction levels are followed very closely and considerable amount of specially designed software modules are being developed.

10. References

Andreasen, Mogens Myrup (2003). *Relations between modularisation and product structuring*. In Proceedings of the 6th workshop on Product Structuring – application of product models, MEK-DTU, Denmark, pp. 1-15.

Berman, B. (2002). Should your firm adopt a mass customization strategy? *Business Horizons*, 45(4):51–60.

Davis, S. (1989). From future perfect: Mass customizing. *Planning Review*.

Du, X., Jiao, J., and Tseng, M. M. (2000). Architecture of product family for mass customization. In Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology.

Felfernig, A., Friedrich, G., and Jannach, D. (2001). Conceptual modeling for configuration of mass-customizable products. *Artificial Intelligence in Engineering*, 15:165–176.

Faltings, Boi and Freuder, Eugene C. (Ed.) (1998): *Configuration - Getting it right*. Special issue of *IEEE Intelligent Systems*. Vol.13, No. 4, July/August 1998.

- Gilmore, J. and Pine, J. (1997). *The four faces of mass customization*. Harvard Business Review 75 (1).
- Hvam, L. (1994). *Anvendelse af produktmodellering, -set ud fra en arbejdsforberedelsessynsvinkel* (in Danish). PhD thesis, Driftteknisk Institut, DTU.
- Hvam, L. (1999). *A procedure for building product models*. Robotics and Computer-Integrated Manufacturing, 15:77–87.
- Jiao, J., Tseng, M. M., Duffy, V. G., and Lin, F. (1998). *Product family modeling for mass customization*. Computers & Industrial Engineering, 35:495–198.
- Jørgensen, Kaj A. (1998): *Information Modelling: foundation, abstraction mechanisms and approach*. In: Journal of Intelligent Manufacturing, vol.9, no.6, 1998. Kluwer Academic Publishers, The Netherlands.
- Jørgensen, Kaj A. (2002): *A Selection of System Concepts*. Aalborg University, Department of Production.
- Jørgensen, Kaj A. (2003): *Information Models Representing Product Families*. Proceedings of 6th Workshop on Product Structuring, 23rd and 24th January 2003, Technical University of Denmark, Dept. of Mechanical Engineering.
- Jørgensen, Kaj A. (2005): *Product Modelling on Multiple Abstraction Levels*. Proceedings of IMCM'05 Klagenfurt, Austria, June 2-3, 2005.
- Jørgensen, Kaj A. (2008): *Product Configuration and Product Family Modelling*. Working Papers, Aalborg University, Dept. of Production.
- Lampel, J. and Mintzberg, H. (1996). *Customizing customization*. Sloan Management Review, 38:21–30.
- Männistö, T.; Peltonen, H.; Soininen, T. and Sulonen, R. (2001): *Multiple Abstraction Levels in Modelling Product Structures*. Data and Knowledge Engineering no.36, pp.55-78, 2001.
- Pine, B. Joseph (1993): *Mass Customization - The New Frontier in Business Competition*. Harvard Business School Press, Boston Massachusetts, 1993.
- Pine, J., Victor, B., and Boyton, A. (1993). *Making mass customization work*. Harvard Business Review 71 (5), 71(5):108–119.
- Reichwald, R., Piller, F. T., and Möslin, K. (2000). *Information as a critical succes factor or: Why even a customized shoe not always fits*. In Proceedings Administrative Sciences Association of Canada, International Federation of Scholarly Associations of Management 2000 Conference.
- Rosch, Eleanor (1978): *Principles of Categorisation*. In: Cognition and Categorization. Laurence Erlbaum, Hillsdale, Jew Jersey, 1978.
- Rumbaugh, J.; Jacobson, I. and Booch, G (1999): *The Unified Modeling Language Reference Manual*. Addison-Wesley 1999.

Sabin, D. and Weigel, R. (1998): *Product Configuration Frameworks - A survey*. In IEEE intelligent systems & their applications, 13(4):42-49, 1998.

Silveira, G. D., Borenstein, D., and Fogliatto, F. S. (2001). *Mass customization: Literature review and research directions*. Int. Journal of Production Economics, 72:1-13.

Skyttner, Lars (2005): *General Systems Theory*. 2nd edition, World Scientific Publ, 2005.

Smith, J. M. and Smith, D. C. P. (1977): *Database Abstractions: Aggregation*. Communications of the ACM vol.20, no.6. pp.405-413 New York 1977.

Smith, J. M. and Smith, D. C. P. (1977): *Database Abstractions: Aggregation and Generalization*. ACM transactions on Data Base Systems, vol.2, no.2. pp.105-133 New York 1977.

Sowa, John F. (1984): *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.