Product Family Modelling for Manufacturing Planning

Jørgensen, Kaj Asbjørn; Petersen, Thomas Ditlev; Nielsen, Kjeld; Habib, Tufail

Published in:
Proceedings of the 21th International Conference on Production Research, ICPR 2011, 1.-4. August, Stuttgart, Germany

Publication date:
2011

Document Version
Early version, also known as pre-print

PRODUCT FAMILY MODELLING FOR MANUFACTURING PLANNING

K. A. Joergensen, T. D. Petersen, K. Nielsen, T. Habib
Aalborg University, Department of Mechanical and Manufacturing Engineering
Fibigerstraede 16, DK-9220 Aalborg, Denmark

Abstract
To enable product configuration of a product family, it is important to develop a model of the selected product family. From such a model, an often performed practice is to make a product configurator from which customers can specify individual products from the family.
To get further utilisation of the product family model, however, the model should be enriched with data for planning and execution of the manufacturing processes. The idea is that, when any individual product is specified using the product configurator, a product model can be extracted with all data necessary for planning of the manufacturing processes. Obviously, data for identification of all used modules and components are included in the product model but also for instance data for processing and assembly operations must be available. These data are not always related entirely to the modules and components but are sometimes also dependent on the specific assembly structure of the configured product, i.e. the combination of modules.
In this paper, issues of how to create manufacturing structures and related planning data in product family models are presented. Primarily, the more complicated multi-level manufacturing structures are regarded and it is argued that the models need to specify other structures for manufacturing compared with the product structure for configuration. Further, the problem of including attributes for of planning data is addressed and it is shown this may be complicated because such specifications may be dependent of the structure and need to be represented in a general for valid for all possible configurations.

Keywords:
Mass customisation, product configuration, product family model, product family modelling, manufacturing structures, manufacturing planning data.

1 INTRODUCTION

Product Configuration and Product Family Modelling have been important topics since Mass Customisation (MC) was adopted more than one decade ago. This research topic was initiated with Davis' publication "From Future Perfect: Mass Customisation" [4], 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" [15], [16], 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 [14], [7], [20], [22], [3].

At many companies, who in the last few years have performed mass production, a demand for customised products is clearly registered. These companies are aware that the market segments cannot be regarded homogeneous; the needs vary from customer to customer and products have to differ equally. So, many companies can find a potential in greater focus on the customers and to use this as a strategic opportunity to establish a competitive advantage.

The combination of mass production and the capability to offer customised products is termed *mass customisation* [15]. As shown in figure 1, the aim is that mass customisation should take the benefits from both one-of-a-kind production and mass production with respect to customisation and production volume. In order to reach this goal, a considerable readiness has to be developed. The products have to be designed for this and the production must be well prepared.

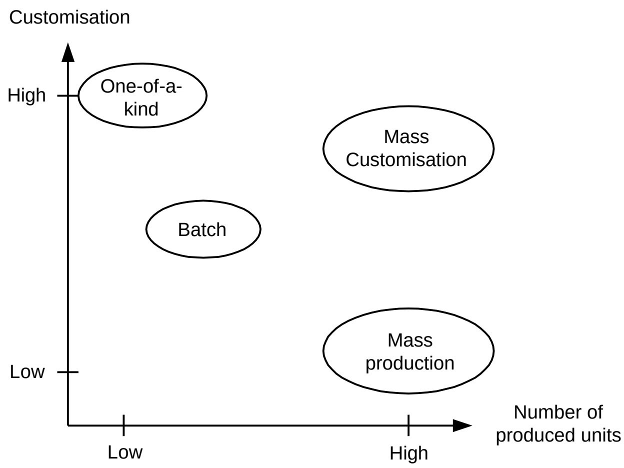

Figure 1: Three production forms positioned regarding customisation and production volume.

Newer research underlines that MC is a strategic non-reversible development and suggests that the change process is considered as a strategic mechanism [21]. Consequently, in order to benefit from MC, the mangers must tailor the development process and relate to the existing business. Many methods for supporting MC focus on modelling the solution space of configuration processes. This means that they do typically not focus on information, which is not directly used to perform the configuration itself. Some of these issues are handled on related areas like cost calculation [27]. This information could include e.g. customer, logistics and manufacturing information [17], [1]. 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 model must be structured in an appropriate way.

The fact that products must be easily customisable in order to achieve MC has been described comprehensively in the literature. [3] and [15] proposed that the use of modular product design [9] combined with postponement of product differentiation would be an enabler to a successful MC implementation. This issue of course also relates to the question of agility of the value chain.

2 PRODUCT FAMILY MODELS

A product family is simply defined as the set of all possible end-products from which the customer can make his selection. A model of a product family, termed the product family model, is then defined as a single model from which models of all end-products of the family can be derived (see figure 2). The product family model can serve as a foundation for the configuration process and, in order to secure that only legal configurations are selected, the model should contain restrictions about what is possible and not possible. The result of each configuration will be a model of the configured product. This model will describe the end-product to a certain degree by a set of specifications, which have to be decided during the configuration process [12].

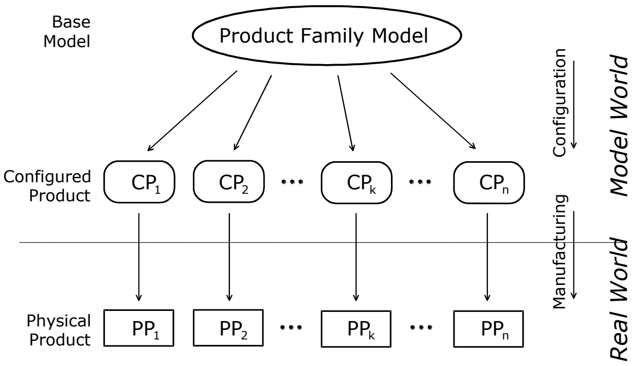

Figure 2: The product family model as the foundation for product configuration.

Ideally, it should be possible to produce the physical product form the model of the configured product (see figure 2). To enable this, each product model must have sufficient specifications in order to determine modules, structure and attributes of the end-product. Consequently, this requires that the product family model is very complete and organised in a way so that the necessary specifications can be generated to each possible configuration.

A product configurator is defined as a tool, computer software, which can support users to perform the configuration process [5], [26]. Often, such configurators are built in a modelling tool and aimed at specific user groups, e.g. sales agents. It is, however, important to aim at the right customisation level and to analyse what different types of users will be engaged. Obviously, it can be a great advantage to build the configurator on the basis of a product family model in which many important characteristics are modelled in advance. However, aiming at different ways of customisation and a variety of user groups is more a matter of creating the right user interface, preferably so that it can adjust dynamically to individual users.

A product family model is basically a computer-based model and such models are fundamentally stored in computers as data objects and data structures, which can be manipulated by applications. Therefore, modelling includes both development of a data model and a number of applications with relationships to the data model [10]. One of the most important requirements for the data model is that it is non-redundant so that no data value is stored more than once. In order to ensure that this requirement is fulfilled, the model representation has to be considered very carefully based on the meaning of data, the semantics. Therefore, the foundation for a data model is an information model ([8] and [19]), created in combination with semantics from the domain, which the design model is addressing.

When an information model is developed, a foundation for the components must be established by creating *types* of model components (see figure 3). These types are the primary content of information models and it is important to distinguish between modelling on the object level and modelling on the type level.

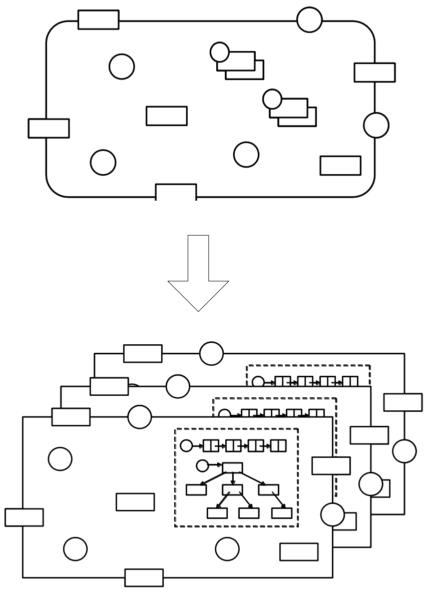

Figure 3: Component types are the basis for objects

3 PRODUCT FAMILY MODELLING

Modelling in various forms has always been a very important approach in design projects and new opportunities with computer-based modelling tools have made it even more important. Such tools have become more useful and with an increasing number of functionalities.

Often, the modelling tools dictate certain modelling methodologies with a number of limitations. However, modelling can be performed in many ways and can have different meanings to designers. The emphasis can be set on many subjects, decisions can be sequenced in many ways and resources can be allocated variously. Companies, who are implementing product configuration, need a comprehensive terminology and a systematic methodology in order to develop their modular products. It is of great importance to use well-defined terms and use the agreed terminology consistently in connection with a well-proven methodology, so that misunderstandings can be avoided and communication can be eased [2].

An important fundamental issue of information modelling is about *abstraction mechanisms*, which provide the means for identification and design of invariant components and structures ([23], [24], [18], [25] and [6]). Two abstraction mechanisms are defined here: composition and classification [11]. 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 5 for a sample computer. Classification focuses on identification of classes/types of components based on the *properties/attributes*, which characterise them. The relationships generalisation versus specialisation are represented by the branches in the taxonomy ([19]).

The two abstraction mechanisms are used in design tasks, but, as indicated in figure 4, 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.

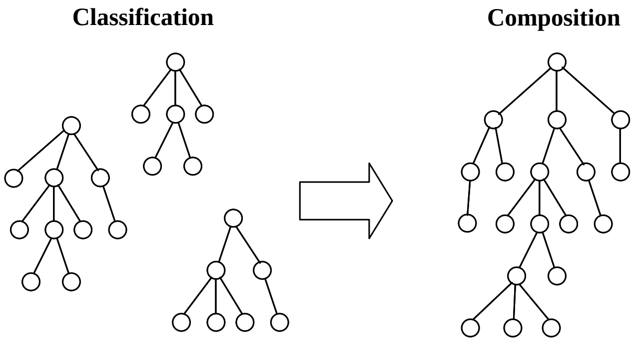

Figure 4: Classification and composition hierarchies

4 REPRESENTATION OF PRODUCT FAMILY MODELS

As stated above, product family models must be able to construct individual product models through a configuration task. Each product model must have sufficient data about attributes and structure to describe and manufacture the physical product. Consequently, the basic elements of product family models are the total set of attributes of the possible product models and the set of identified modules, each with their internal attributes and data structures.

In a previously published simple modelling language [13], the generic representation is a component type as shown in the following examples:

```
type Computer
{
    Prize : Number;
    OperatingSystem : Boolean default true;
    Colour = Case.Colour;
    HardDisks = HardDisk;
    DiskMemory = Sum(HardDisk.StorageCapacity);
    Contents ->>
    { 1..1 Case; 1..* Keyboard; 0..* Mouse; 1..2 Monitor; }
    constraints
    {
        Monitor <= 2;
        HardDisk + CdDrive + DvdDrive  <=  DiskCable * 2;
        OperatingSystem ⇒
            HardDisk.OperatingSystem <> Non;
        CdDrive not ⇔ DvdDrive;
        .....
    }
    .....
}
```

```
type HardDisk
{
    Name :          string(50);
    StorageCapacity :       integer;
    AccessTime : float;
    Price : currency;
    PreSet : {Master, Slave} default Master;
    OperatingSystem :
    {Non, WinXP, Win2000, WinMe} default WinXP;
    .....
}
```

As indicated, a configurable computer is used as illustration. The examples show attributes with data types or domain values and optional default values. Attributes can be a function of other attributes in the same type or in other types. This can be modelled by an *expression* with standard functions or special functions as a special algorithm. If the name of a module type is included in such an expression, it means “number of instances of the type”. The symbol ->> represents a collection, which is a basic means for representation of structures. Collections may be specified with multiplicity, e.g. 1..* denoting one-to-many. Constraints can be formulated as arithmetic or logical expressions. Here, the ordinary arithmetic operators like addition, subtraction, multiplication and division can be used together with standard functions. The following arithmetic relationship operators =, >=, <=, >, < and <> can also be used along with the logical operators AND, OR, XOR, NOT, implication (⇒) and bi-implication (⇔). If the name of a module type is included in a logical expression, it means ”instance of the type”.

5 MODELLING FOR MANUFACTURING

When a product family model is represented as an information model, all possible data can be included. Often, the models focus on data, which are necessary only to perform the configuration process and provide an overview over the resulting product structure like the one in figure 5. Sometimes, additional data are presented, e.g. selected attributes of the product or even graphic data, which can be shown in viewers.

Product structure:

- Computer
  - Case
    - Cpu
      - Cpu Board
      - Processor x 2
      - Memory Unit x 3
      - Graphic Board
      - Sound Board
    - Mass Storage
      - Hard Disc
      - DVD Drive
    - Power Supply
    - Lock
  - Keyboard
  - Mouse
  - Monitor
  - Power Cable

Figure 5: Sample composition structure of a computer

Obviously, product family models can potentially be utilised much more, for instance as the foundation for manufacturing. This means that, when a product has been configured, the specific model for this configuration should have valuable data for efficient and effective planning and execution of manufacturing tasks. According to figure 2, the manufacturing data should be included in the product family model and carried over to the product model as a result of the configuration. In the following, these considerations are only limited to planning data but similar results can probably be shown for other applications.

Based on such requirements for the product family model, a number of modelling issues must be considered. In the simplest form, data may be added to the existing components of the model and easily presented from these. Often, however, additional components must be created and it may be necessary to form new structures of the model. Consequently, the information modelling process may be more complicated.

5.1 Manufacturing Structures

As already mentioned, the product structure is an often occurred description of a product and usually this structure is shown a result of configuration (see figure 5). However, such a structure can be created in many ways dependant on the purpose. This means that structures, which are suitable for users involved in configuration (by use of a configurator) may not be useful for manufacturing. Even in this context, multiple structures may be preferred.

For operational planning, the primary focus is on the flow of components and the operations performed on the components. To describe this, a rather deep tree structure may appear because each branch will represent an operation, where one or more components are taken as input and a new component is delivered as the output. For each input item, a quantity is specified. In the following, this general view is simplified so that a number of operations are collected into one operation (see figure 6).

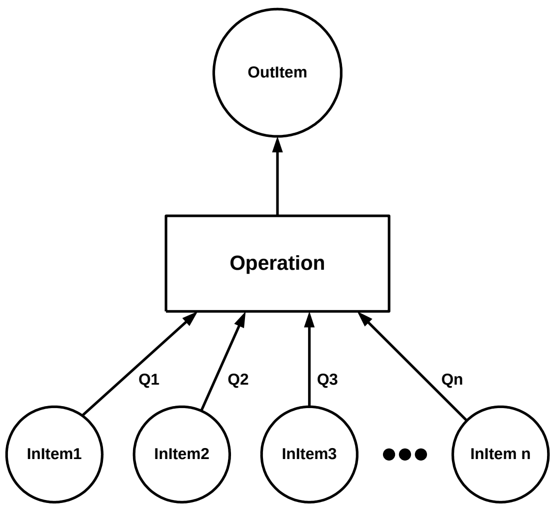

Figure 6: General structure for manufacturing

In connection with a larger product structure, this general structure can be applied to all levels, i.e. from components to components, from components to modules, from modules to modules and from modules to end-products.

For the computer example, a suitable manufacturing structure different from the one in figure 5 could be the structure shown in figure 7. The operations are not shown but they are assumed at each transition from one level to another. The example shows only the major assembly structure and not many of the underlying sub-components and the operations. On the other hand, the assembly operations are normally the primary operations in connection with product configurations.

Manufacturing structure:

- Stationary Computer Cx20
    - Case
        - Case Basis
            - Case Frame
            - Cpu Board
            - Power Supply
            - Lock
        - Processor x 2
        - Memory Unit x 3
        - Graphic Board
        - Sound Board
        - Hard Disc
        - DVD Drive
        - Panel x 2
    - Keyboard
    - Mouse
    - Monitor
    - Power Cable
    - ....

Figure 7: Sample composition structure for assembly

Design of the manufacturing structure may involve decisions about important issues like modularity, platforms and postponement and how they should be handled in connection with configuration. With reference to figure 7, for instance, the Case Basis could be characterised as a platform, which could be used equally for many (perhaps all) computers and print boards are good examples of modules. The processors, memory units, print boards and mass storage modules are often of a type, which are selected differently by customers in the configuration process and, consequently, they will be candidates for postponement and are placed on the same level late in the assembly operations. In contrast, the lock is assembled on a lower level and should perhaps be moved upwards.

To make it possible to generate the manufacturing structure and perhaps other structures for all possible configurations and transfer it to the model of the configured product requires that the product family model must have the necessary data and specifications. Structures can be described by collections and for the computer example, the computer case collection suitable for configuration is

Contents ->>
{ CPU; MassStorage; PowerSupply; PowerCable; }

The corresponding collection describing the manufacturing structure similar to the one for the computer case in figure 7 will then look like this

Manufacture ->>
{CaseBasis; Processor; MemoryUnit; PrintBoard; MassStorage; }

Observe that it is not necessary to specify the collection further by including multiplicity clauses because this will be defined by the configuration constraints.

5.2 Manufacturing Planning

Data for manufacturing planning may be represented in different ways but most often, they are included in the component types as attributes and, for each specific configuration, they can be retrieved by use of the manufacturing structure. An obvious example is to create a complete overview over all the components included in the product and for each branch, these components are input to and output from the operation (see figure 6). If cost prices attribute are available in each component, the total material cost can be calculated. Also, to each component, additional data like required extra materials (ex. screws), equipment (ex. screw driver), labour, and operation time can be specified. The same kind of data can be related to the operations and represent values, which are independent of the input components. These attributes can be included in output component. Altogether, the total assembly operation time for the computer case may be formulated like this

```
Case.TotalOpTime = Case.AddOpTime +
Manufacture.Aggregate(OpTime);
```

Observe that for each specific configuration, only the operation time values for the included components will be aggregated.

Sometimes, it may be necessary to add constraints, which specify certain cross-going relationships between components. For instance, a specific assembly sequence may be required. Such requirements must also be represented in a general form, which can be handled for all possible configurations. In case of the sequencing example, a priority attribute could be added to the component types and then the manufacturing collection could be iterated according to the values of this attribute. Another special example could illustrate a relationship between components on different levels of the manufacturing structure. If for instance, certain components require specific test procedures after the assembly operations have been carried out, the operation time for the referred output component will be increased. Such relationships are much more difficult to model in product family models because they have to be represented in a more general form.

As stated, product family models are used for developing product configurators and the result of configurations is often just a simple list of the included components. Based on a product family model with much more data aimed for manufacturing planning, much more precise structures and planning data estimates may be provided and included in models of the configured end-products. Typically, such data are submitted to the Enterprise Resource Planning (ERP) system for generation of inventory requests and job schedules. However, if there is a clear distinction between components, which are manufactured independent of configuration, then the manufacturing planning of these components are traditional an suitable for the ERP system. In contrast, planning of the manufacturing – often just the assembly operations – may be handled much more suitable by the configurator. In this case, it would be obvious to separate the planning functionalities from the configuration functionalities and perhaps perform the configuration on two steps; first the sales configuration and afterwards the technical configuration with the manufacturing planning. Of cause, exchange of data between the configurator and the ERP system will be necessary.

6 CONCLUSION

The increasing demands from the customers to customisation have led to the introduction of mass customisation and product configuration. Most often, the introduction and implementation of product configuration is only aimed at product sales and not utilised sufficiently for other influenced activities.

Mass customisation and product configuration demand a systematic way of thinking in design, manufacturing, and maintenance of the configurable products. This can be achieved by developing a product family model as a model of the complete set of possible end-products. Such models include description of component types with attributes, domains, structures and constraints. The constraints are used to describe the different relationships, dependencies and connections between the module types and their attributes. They are important because many relationships have to be represented in a rather general form in order to be valid for all possible configurations.

If product configuration should be utilised much more for manufacturing of the individually configured end-products, a valuable basis could be established by extending the product family models. To do so, it is necessary to include a specification of a manufacturing structure and additional attributes with different kinds of planning data. It must be realised that the manufacturing structure – the structure of components organised for the manufacturing operations – most often will be different from the structure, which is most suitable to present for the sales configuration. Consequently, it is necessary to specify multiple structures in product family models. Furthermore, attributes for planning data must often be specified with a dependency to the manufacturing structure and this leads to a higher degree of complexity when developing product family models. Some examples of these issues are stated and illustrated based on a simple computer family model.

7 REFERENCES

[1] Abdi, R. et al. (2006). Reconfigurable Manufacturing Systems and Transformable Factories. Ed. Anatoli I. Dashchenko. Springer Berlin Heidelberg New York.

[2] Bellgran, Monica and Säfsten, Kristina (2010): Production Development - Design and Operation of Production Systems. s.l. : Springer-Verlag London.

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

[4] Davis, S. (1989). From future perfect: Mass customizing. Planning Review.

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

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

[7] Gilmore, J. and Pine, J. (1997). The four faces of mass customization. Harvard Business Review 75 (1).

[8] Hammer, Michael and McLeod, Dennis: *The Semantic Data Model: A Modelling Mechanism for Data Base Applications.* Proceedings of ACM/SIGMOD International Conference on Management of Data. Austin Texas, pp.144-156, 1978.

[9] Harlou, Ulf (2006): Developing product families based on architectures - Contribution to a theory of product families. s.l. : Department of Mechanical Engineering - Technical University of Denmark, 2006.

[10] M. Jazayeri, A. Ran and F. van den Linden: Software architecture for product families: Principles and practice. Addison-Wesley, 2000.

[11] Jørgensen, Kaj A.: Information Modelling: foundation, abstraction mechanisms and approach. In: Journal of Intelligent Manufacturing, vol.9, no.6, 1998. Kluwer Academic Publishers, The Netherlands.

[12] Jørgensen, Kaj A.: 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.

[13] Jørgensen, Kaj A.: Product Family Modelling - Working with multiple abstraction levels. In: "Advances in Mass Customization and Personalization, Vol. 3". World Scientific Publishing Co, USA, 2009.

[14] Lampel, J. and Mintzberg, H. (1996). Customizing customization. Sloan Management Review, 38:21–30.

[15] B. Joseph Pine: Mass Customization - The New Frontier in Business Competition. Harvard Business School Press, Boston Massachusetts, 1993.

[16] Pine, J., Victor, B., and Boyton, A. (1993). Making mass customization work. Harvard Business Review 71 (5), 71(5):108–119.

[17] Reichwald, R., Piller, F. T., and M¨oslein, 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.

[18] Rosch, Eleanor: Principles of Categorisation. In: Cognition and Categorization. Laurence Erlbaum, Hillsdale, Jew Jersey, 1978.

[19] Rumbaugh, J.; Jacobson, I. and Booch, G: The Unified Modeling Language Reference Manual. Addison-Wesley 1999.

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

[21] Salvador, F., de Holan, P. M., Piller, F. (2009). Cracking the Code of Mass Costomization. MITSloan Management Review. Vol 50, no. 3, Spring 2009.

[22] 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.

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

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

[25] Sowa, John F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, 1984.

[26] Yu, Bei and Skovgaard, Hans Jørgen: A Configuration Tool to Increase Product Competitiveness. In IEEE Intelligent Systems, vol. 13, no. 4, 1998.

[27] Zhang, M., & Tseng, M. M. (2007). A Product and Process Modeling Based Approach to Study Cost Implications of Product Variety in Mass Customization. IEEE Transactions on Engineering Management, 54(1).