DTU Library

Proceedings from the International Conference on Economic, Technical and Organisational aspects of Product Configuration Systems

Edwards, Kasper; Hvam, Lars; Møldrup, Morten; Møller, Niels; Riis, Jesper; Pedersen, Jørgen Lindgaard

[Link back to DTU Orbit](#)

International Conference on Economic, Technical and Organisational aspects of Product Configuration Systems

Technical University of Denmark
June 28-29th 2004

Conference Proceedings

Proceedings from the International Conference on Economic, Technical and Organisational aspects of Product Configuration Systems

Organizing Committee

The International Conference on Economic, Technical and Organisational aspects of Product Configuration Systems has been organized by

- Assistant Professor Kasper Edwards,
- Associate Professor Lars Hvam,
- Ph.D. Student Morten Møldrup,
- Associate Professor Niels Møller,
- Associate Professor Jørgen Lindgaard Pedersen and
- Assistant Professor Jesper Riis.

Scientific Committee

- Professor Cipriano Forza, University of Modena & Reggio Emilia, Department of Mechanical and Civil Engineering, Italy
- Professor Bo Helgeson, Work Practice Laboratory, Dept. of Human Work Science and Media Technology, Blekinge Institute of Technology, Sweden
- Dr. Lars Hvam, Department of Manufacturing Engineering and Management, Technical University of Denmark
- Senior researcher Håkon Finne, STEP, Center for innovation research, SINTEF, Norway
- Docent Staffan Laestadius, Department of Industrial Economcis and Management, Royal Institute of Technology, Stockholm, Sweden
- Professor A. Pennathur, Mechanical and Engineering Department, University of Texas at El Paso, U.S.A.
- Dr. Frank T. Piller, Director of Tum research Center Mass Customization, TUM Business School, Technische Universitaet Muenchen, Germany
- Dipl.-Ing. Ralph Seelmann-Eggebert, Fraunhofer Institute for Factory Operation and Automation, Information, Logistics and Automation Systems, Germany
- D.Sc. (Tech), Professor Timo Soininen, Software Business and Engineering Institute, Helsinki University of Technology, Finland

Acknowledgement

Table of Contents

TECHNOLOGY REVIEW OF MASS CUSTOMIZATION

Karthik Ramani, Robert Cunningham, Srikanth Devanathan, Jayanti Subramaniam, Harshal Patwardhan
PRECISE[1]School of Mechanical Engineering
Purdue University
585 Purdue Mall
Mechanical Engineering Bldg.
West Lafayette, IN 47907
Email: ramani@purdue.edu

Abstract: This paper provides a comprehensive review of how technology has and is still aiding the development of product customization to achieve mass production costs while being responsive. With the technological advances of manufacturing, information technology, communication, and product design, higher levels of mass customization can be realized. Industry examples are used in categorization of mass customization and corresponding technology is presented. In addition, key technological advances that is enabling mass customization are provided. The overall technology assessment is performed.

1. INTRODUCTION

Just as mass production was crucial to manufacturing in the 20th century, mass customization (MC) will be the key to economic growth in the 21st century. MC is the ability to design and manufacture customized products tailored to meet a customers needs at mass production costs and speed (see Figure 1). While organizations continue to outsource for economical reasons, managing the interfaces between suppliers has become expensive and inefficient. By dispersing engineering and production geographically, manufacturers have increased the number of places their knowledge interfaces and multi-tier supply chains can breakdown. This amasses hidden costs, increases lead times, and reduces control. This is especially true for customized products that have tight deadlines.

Significant reduction in inventory costs and lower obsolescence will also occur in a manufacture-to-order environment. The survival and growth of all companies, small to large alike will then be dictated by their ability to cater to this emerging market trend. Companies such as Dell, 80/20 Inc., and Lutron have clearly proven that customers prefer this type of sales model. This model requires manufacturers to be market-driven and customer-responsive, which means offering more product variation and allowing customization. However, adopting this model poses some serious challenges for traditional manufacturing environments. It dramatically complicates the manufacturing system design if the same design procedures that have been developed for standard products are used. In the MC model, value is created by people designing the product, combining the supplier competencies, establishing supplier networks, and ensuring customer satisfaction.

Figure 1. From Mass Production to Mass Customization

Among the many challenges for MC, the following are important: (i) keeping costs low to match those of standardized items, (ii) achieve high quality production of high variety of products, (iii) making these products available in a timely fashion to customers. A series of industrial shifts enable MC: (a) Modularization of products and processes has enabled management of product variety: and (b) The ability of "knowledge-based" software to configure products and (c) Flexible automation for manufacturing has been enabled because of improved low cost technologies. Hence, today's manufacturing systems have the potential to build a large variety of end products at costs comparable to mass-produced items. However,

---

[1] PRECISE; Purdue Research Education Center for Information Systems in Engineering

this potential is just the beginning to be realized based on the complexity of the product, manufacturing, and supply chain. Therefore, different manufacturing sectors have different business drivers and are at varying degrees of readiness to adopt MC methodologies.

With these trends, many issues arise in the product development and production cycle. These issues are being addressed by capabilities in computational, communicational, and informational areas creating innovations in flexible automation, networks, and electronic product design. An increasing number of companies are adopting mass customization strategies at different levels in their product development cycles (see Figure 2). In this review paper we seek to address some of technological innovations that aid in implementing MC. A case study is also presented for demonstrating a knowledge-based product configurator.
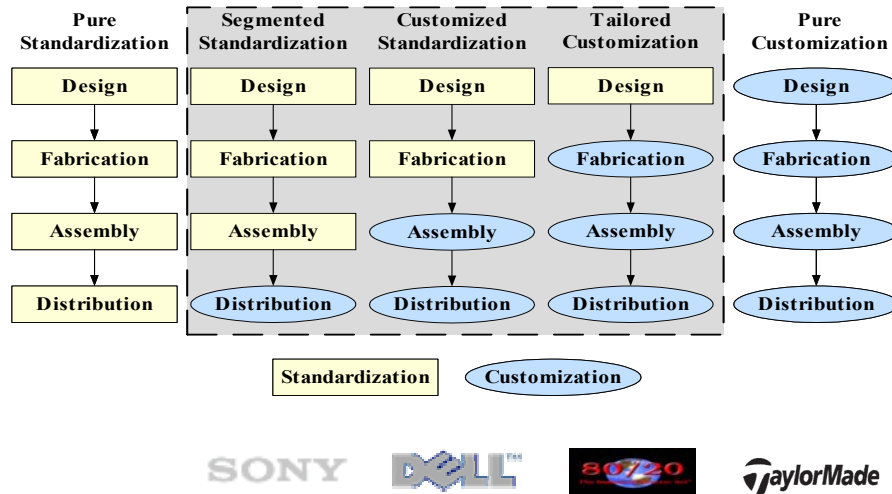
Figure 2. Levels of Customization (adapted from [Lampel96])

2. PRODUCT DESIGN AND DEVELOPMENT

2.1 Modularity and Flexibility

One of the difficulties with mass customization is that product variety increases drastically with just a few product options. This will likely stretch a company's own infrastructure, especially with regard to variety of tooling and fixtures needed to manufacture a wide range of product variants. Also, part numbers rise, adding complexity to logistics and manufacturing processes. As a result, there is increased outsourcing of significant portions of complex products such as automotives which allows more product derivatives to be built in their own facility without having to invest in expansion of their own plants. However, outsourcing alone may not solve the high part count problem unless the product and interface definitions are standardized across the supply chain. In other words, companies need tools to manage the product rather than product data.

One of the solutions to managing the complex variety is the product platform which is often defined by means of the product architecture. The product platform has been defined by McGrath [McGrath95, Halman03] as a set of subsystems and interfaces that form a common structure from which a stream of related products can be developed and produced efficiently. Baldwin and Clark [Baldwin97] define three aspects of the underlying logic of a product platform: (1) its modular architecture; (2) the interfaces (the scheme by which the modules interact and communicate); and (3) the standards (the design rules to which the modules conform).

Standardization and flexibility of products and processes is an important part of any mass customization initiative. This is achieved primarily through modularization of products, processes, and the supply chain. Modularization demands standardization of product interfaces and allows higher variety and customization. Trends towards modularization of products and increased use of product-platform approach are being observed in many manufacturing firms such as Sony, Siemens, and Lutron. An increasing number of automotive OEMs are consolidating and updating their assembly plants so that each of their remaining production facilities can produce a higher annual output and a wider variety of body styles built upon modular underbody structures. This is, in part, due to the increasing market pressure on OEMs to produce more niche vehicles. This in turn demands a wider introduction of flexible manufacturing systems into their body shops capable of building multiple body styles through the same production facility.

Another important aspect of today's product development process is the convergence of multiple disciplines and technologies in providing products with higher efficiency, better safety, and more customization. The automotive industry has had one of the biggest impacts due to this trend. It is estimated that almost 25% of the base cost of an automobile is due to the electronics and software. For the first time, technology has reached a point where it can be expected to provide consistently, reliably, and cost-efficiently the systems and software that will allow a level of cohesive, networked communication throughout a vehicle. This benefits safety, performance, emissions reduction, and comfort [ref] for the automobile. A convergence of various technologies in different disciplines such as electronics, mechatronics, materials and manufacturing systems has lead to improved products and better customer satisfaction.

Interesting issues arise while developing modular components and assemblies, especially with regard to interfaces between components. The relationships between the various modules will change dynamically whenever the product, process, or distribution changes. This warrants a different organization of the communication and information infrastructure. These include both the physical interfaces between modules such as geometric interference and tolerances, as well as software interfaces. In addition, for products driven by electronics the communication and information interfaces between the different modules will need to be standardized and well-defined. From a product design perspective, Computer Aided Design (CAD) and Computer Aided Manufacturing systems have become mature and allow third-party interfacing and applications. Several standards for information communication, such as XML and STEP, have emerged in the past decade without significant impact in the manufactured products domain. Several software solutions in the area of Product Lifecycle Management (PLM) and Product Data Management (PDM) have emerged in the past decade. PDM/PLM systems are being used by various manufacturers to provide workflow and store/manage product-process data. PLM/PDM systems (1) offer no significant value addition to the core product, (2) do not spot design errors, and (3) are not product-centric (but data-centric). PDM/PLM systems merely manage the data through functions once done by other slow modes of interaction across distances. They do not add inherent value to the product itself.

In contrast, a transition to mass customization needs a "product centric" model that has the following:
1. Flexible product family and supply chain modularity,
2. Faster design tools,
3. Quality designed into the part,
4. Optimization of engineering decisions, and
5. Immediate part procurement (reuse).

2.2 Design and Product Configuration

Design tools have been aided by the technological innovations. Some of the most widely used design tools are Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), and Computer Aided Engineering (CAE). CAD is used as a means to link a part model with analysis software (CAE), quickly create machining code and visual simulation information (CAM), and provide a form of communication (usually visual) for the product.

The major CAD innovation that provides an enabling step to mass customization is parametric modeling techniques. Parametric modeling essentially creates flexible product models that can easily be modified geometrically. This geometric modification can also be linked to the analysis software packages such as finite element analysis (FEA), computational fluid dynamics (CFD), user defined analysis programs, and more.

The general direction of industry is to perform engineering analysis earlier in the design and to create robust, flexible models. This allows designers to more easily vary designs and see what will happen to a product and its performance if different aspects should be varied, and what could cause failure. To get such answers,

Figure 3 Modularity Integration loop for the cycle of product evolution

[designers] need to put the engineering back into design and turn to CAE [Ell04].

Other companies such as Engineous Software, Inc. link analysis and CAD software with their optimization software, iSight™. This dynamic link is made possible by flexible models and Open APIs. Open application programming interface (API) allows CAD users to create highly flexible models in programming code. It also permits analysis developers to build tools into CAD software. This in turn promotes a cyclical product evolution from technological innovation (see figure 3).

Since CAD software is now generally needed for product development, trillions of 3D CAD models have been created.

In fact, CAD model libraries have been formed as parts catalogs. The designer can now easily insert a CAD model of a part that will be used in the product from the supplier instead of having to create the part model from scratch. The latest version of Pro/Engineer Wildfire ™ CAD program incorporates a direct link with CADregister™, a CAD parts catalog run by ThomasRegister™. The user can drag and drop parts directly into the CAD software environment for analysis and representation. Other CAD software companies such as SolidWorks™ and Unigraphics™ are doing the same while also allowing user defined parts libraries. This saves the designer a significant amount of time in terms of procurement, modeling, and analysis.

In order to manage and keep track of the parts and products within a company, further software suites were developed. Product Lifecycle Management (PLM) software is intended to aid the process of design to production by managing the information attached to products, suppliers, processes, and resources. Enterprise Resource Management (ERP) software aids upper management in getting an overall view of operations regarding the product.

Recently, a new set of design solutions, called Product Configurators (PC), have become significant in addressing many of the design issues related to mass customization. Design is the most important piece of information that describes and defines a product. Product Configurators focus on the product design itself, and attach other process and product information to the design, thereby taking an integrated (product-centric) approach to product representation.

Configuration is "*... the construction of a physical system according to specifications by selecting, parameterizing, positioning and assembling instances of suitable existing component types from a given catalog"* [Gunter 99].

Product Configurators enable: (1) Rapid product development (2) Design reuse, (3) Reduced design time, (4) Collocation of parts in viable design, (5) Engineering rules-based design, and (6) Enhanced product reliability. In summary, product configurators in conjunction with CAD/CAM systems and flexible automation (e-Factories of the future) have the potential to achieve the goals of data management systems with regards to rapid product development. These product configurators have emerged as the newest design tools for 21st century product development and will play a key role in realizing the goals of mass customization. In addition, new systems for managing product information enable reuse of past designs, and thus reduce product development time [Iyer03, Lou04].

Since CAD is essentially the design portal for products, software that directly translates customer requirements into design concepts in the CAD system is necessary. Product configurators fill this need to link the customer to the design stage. Designers or end customers use product configurators to create a product from a set of predefined options or variables. The level of configuration depends in part to the level of product modularity and maturity. Configurators range from simple tools with limited options to complex rules based systems that bring together all of the parts, products, and processes to meet the customer specifications. We present an industry example of a rules based configurator that demonstrates the customer-to-design bridge.

Case Study from Imaginestics, LLC

The Barker Company designs a product line of merchandiser (display cases) for stores to showcase their products. It was decided that the order-entry process currently performed in MRP, which is Symix, was to be integrated to capture the job number and customer directly. Barker elected to use the BMD series product line for the initial implementation, this happened to be one of their more complex configurations. In addition, it was a high volume product line so any cycle time improvements would be realized substantially. The BMD Series is a full-service merchandiser standard with flat or curved front lift glass and optional mezzanine shelving. This flexible merchandiser allows customers to display their product on one or two rows of easily removable painted metal shelves, or to create a product design on a full single deck (see Fig. 4).

Imaginestics, LLC, developed a customizable rules matrix that allows a company to define their unique environment and special allowable customer configurations. The rules matrix forms the basis for their configuration engine called the i-config engine. These rules are established in Microsoft Access, since most companies are familiar with it. The matrix also gives companies the greatest amount of flexibility as their business and manufacturing environment changes. However, the i-config data repository can be either Oracle or MS SQL. At Barker, three rules matrices were designed to define the product and unique customer configurations, namely, the Parameter Rules matrix, Assembly Rules matrix, and a Material Rules matrix.
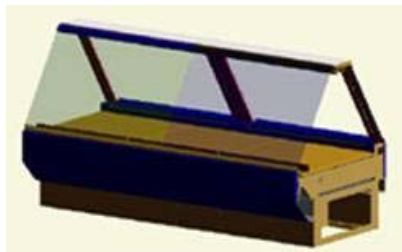
Figure 4 Barker Company Merchandiser configuration (Courtesy: Imaginestics, LLC)

Parameter Rules Matrix - Parameter rules are broken into control (global) and local variables. Control variables are parameters such as length, depth, height, which are global in nature and affect the overall dimension of the product. Local variables are more specific to a component, such as a type of finish, which defines the thickness variable. Any allowable size configurations or limitations were established for each

assembly and component options in this matrix. All available assemblies and their respective component and feature options were defined in this matrix.

Assembly/Component Relationship Rules Matrix - The assembly and component rules were established in this matrix. The assembly and the available components were entered in an indented bill-of-material type of a chart. Each assembly configuration contains all available subassemblies and their respective components.

Material Rules Matrix - The properties for all available materials and the type of finish were established in this matrix, along with the component or feature that the user had options to select. Local variables such as thickness of the finish and material density were assigned in this matrix.

Parallel to the definition of the rules matrix, solid models were created in CAD for the BMD series. The CAD solid parts were designed using sketches and the sketch entities defining the control (global) variables and local variables were given unique variable names and linked throughout the assembly-component design, using the rules matrix definitions. All available components and their related drafting files were modeled using Solid Edge. Meta-data information was also linked to title block and revision block fields in drafting files, such as job number, designer, customer & build date.

Prior to the rules-based configurator, the solid modeling process at Barker was performed only by the engineering design group and was one step removed from the entire manufacturing information flow. Following the use of the product configurator, not only has the task of creating solid models of product variants been simplified and automated, but has also helped manage the process of converting diverse customer requirements into deliverable final products. This enhanced process has the capability to seamlessly integrate the engineering and design function into the entire product life cycle - customer service to final manufacturing and assembly.

3. FLEXIBLE AUTOMATION

In a similar way that CAD allows for flexible parts and products, software tools have been used to design flexible factories. Also, simulation and analysis can be performed in order to optimize the processes involved in product fabrication and delivery. This software has been generally named virtual factory software. It could also fall under the umbrella of manufacturing process management.

Use of Automation for Cost Competitiveness

To cater to the demands of custom products, companies will require manufacturing systems that have a great deal of agility to respond to changing market needs [ABr97]. A "responsive" manufacturing system is one that can quickly reconfigure itself to allow flexibility not only in producing new products but also in changing the system itself. Such systems will necessitate the use of highly sophisticated automation systems that are flexible, extensible and re-usable. The elements of an Automation System include the following:

a) Manufacturing Systems (refers to actual machines, their flexibility,)
b) Software Support Systems
c) Industrial Communication Systems (refers to systems that enable supply chain integration)
d) Mechatronics
e) Production Control Systems (refers to use of JIT, Lean Manufacturing concepts, six-sigma)
f) E-Business

The maturity of a company in each of these elements determines its overall automation level. The Level of Automation (LOA) in any manufacturing facility plays a very important role in determining the manufacturing effort (i.e. the time required) required to produce a particular part. The LOA also decides the percentage of tasks that will be done using manual labor and the percentage of tasks that will be completed using machines. This breakup determines the net resource rates and consequently decides the cost structure for that company. Other important aspects that impact the automation-cost relationship are the following [AHW95]:

1) Reduced Labor content
2) Increased flexibility
3) Quality / Yield improvements
4) Capacity Increase (leading to quick response)
5) Reduced changeover and installation times
6) Reduced floor space requirements
7) Reduced downtimes
8) Improved Safety

In the past, the prohibitive costs of the automation systems presented a major obstacle for use of automation systems. However, the costs for automation systems have reduced and the above mentioned benefits provide major incentives for companies to adopt automation. With greatly reduced payback period on these sophisticated systems, use of automation is fast becoming a major competitive advantage especially in countries with high labor costs. World leaders in automation

technologies like ABB, Siemens and Rockwell are pushing the envelope for the next generation of factories. Investments in vision systems, sensors, multi-axis and multi-function machines, cells, robotics, Radio Frequency Identification (RFID) tages, wireless systems, simulation, and other technologies are a testimony towards the vision. A "digital factory" which can be designed and simulated on the computer is the ultimate dream of every factory designer and automation specialist [Wuc03]. A "flexible factory" producing "flexible products" will enable increased productivity and competitiveness.

4. SUPPLY CHAIN MANAGEMENT

One of the biggest hurdles faced by companies trying to adopt mass customization has been the lack of flexibility and responsiveness in the supply chain. The supply-chain management research has generally focused on analyzing supply-chain configurations for satisfying the demand for standardized products. A traditional supply-chain configuration consists of a fixed network of raw material suppliers, manufacturers of finished products, distributors of finished products, and the end customer. Issues such as where the components should be procured from, where they should be assembled, how it is possible to minimize the procurement and assembly costs, and what are cost effective ways of distributing these finished products have been addressed for standardized products which are manufactured in high volume. However, increased product variety due to mass customization has led to additional challenges in managing this new dynamic demand chain. Fisher et al. [Fisher99] have shown that the increased product variety, although beneficial to consumers, is making it more difficult for manufacturers and retailers to predict which goods will sell and to plan production accordingly. Traditional production planning systems such as JIT, MRP, and quick response are not able to handle demand chains with customized products. The primary source of difficulty is that the supply chain is not responsive enough to changing demand environments. Even a flexible system, such as the one developed by Dell Computers, is constrained by component suppliers' long lead-times. Pine et al. [Pine93, Pine97] discuss the challenges associated with making mass customization work.

A number of recent studies explore the impact of increasing product variety on manufacturing efficiencies. For example, Fisher et al. show that day-to-day variability in option contents of an automobile assembly has a significant adverse impact on manufacturing efficiencies. However, if the process is optimally buffered, then product variety has an insignificant impact on direct assembly labor. Fisher et al. address issues related to component sharing to manage product variety. Randall and Taylor [Randall01] study the impact of product variety on the supply chain structure in the bicycle industry. Their analysis suggests that firms producing locally (in the U.S.) are more efficient in offering higher product variety than firms off shoring.

To cope with the pressures of competition and product customization, manufacturers have tried to mimic the mass customization ideal of "build-to-order" with "locate-to-order". Here, a product that is similar to the customer's order is located within the companies vast inventory and is shipped. This is a compromise for customers. New trends have emerged where the products are being customized as they are in the Supply chain. This is illustrated by Toyota's Scion where customization is carried out in the Port and at the dealer to achieve shorted order-to-delivery time. [SAE May article]

5. CONCLUSIONS

Therefore, the technological innovations in key areas of design, configuration, automation, and supply chain management are enabling mass customization in today's highly productive companies. The modularity of products coupled with flexible design systems such as CAD, configurators, and automation are creating customized products at mass production costs and high responsiveness. Individual innovations in these areas will lead to higher levels of mass customization. Incorporating these technologies is creating more customer-centric businesses that are able to meet the individual needs of its customers. The next innovations will bring tighter coupling of the customer to product design, supply, and delivery. Our research at PRECISE is focused on customer driven networks and product development. This review provides a foundation for further research.

6. ACKNOWLEDGEMENTS

7. REFERENCES

1. [ABr97]   Allen-Bradley   White   Paper,   "Perspectives   on   future   of   Automation   Control",
   http://www.ab.com/events/choices/

2. [AHW95] Adler, D., Herkamp, J., Wiesler, J., Williams, S., 1995, "Life Cycle Cost and benefits of process automation in bulk pharmaceuticals", ISA Transactions, 34, pg 133-139

3. [Wuc03] Wucherer, I.K., June/July 2003, "The future of factory automation", IEE Computing and Control Engineering, pg 30-35

4. [Ell04] Eliot, L., June 2004, "Which Comes First – CAD or CAE?," Desktop Engineering, pg 10

5. [Lampel96] Lampel, J. and Mintzberg, H. (1996) "Customizing Customization," Sloan Management Review, Vol. 37, pages 21-30.

6. [Pine93, Pine97] [Pine93] Pine, J B., Victor, B., and Boynton, A. C., (1993), "Making Mass Customization Work," Harvard Business Review, September-October 1993.

7. [Pine97] Pine, J B., and Gilmore J. H., (1997), "The Four Faces of Mass Customization," Harvard Business Review, January-February 1997.

8. [Randall01] Randall, T., and Ulrich, K. T, "Product Variety, Supply Chain Structure, and Firm Performance: Analysis of the U.S. Bicycle Industry," Management Science, Vol. 47, No. 12, December 2001 pages 1588–1604.

9. [Fisher99] [Fisher99] Fisher, M., Hammond, J., Obermeyer, W., and Raman, A., (1999), "Making Supply Meet Demand in an Uncertain World," Harvard Business Review, May-June 1994. pages 83-93.

10. [SAE May article]

11. [Gunter99] Gunter, A., and Kuhn, C., (1999), "Knowledge Based Systems - Survey and Future Directions," 5th Biannual German Conference on Knowledge Based Systems, Proceedings Springer, Lecture Notes in Artificial Intelligence 1570, 1999.

12. [Halman03] Halman, J. I. M., Hofer, A. P., and van Vuuren, W., (2003), "Platform-Driven Development of Product Families: Linking Theory with Practice," Journal of Production Innovation Management, Vol. 20, pp. 149–162.

13. [Baldwin97] Baldwin, C.Y. and Clark, K.B., (1997), "Managing in an age of modularity," Harvard Business Review, Vol. 75, No. 5, pp. 84–93.

14. [McGrath95] McGrath, M.E., (1995), Product Strategy for High-Technology Companies. Homewood, IL, Irwin Publishers.

15. [Iyer03] Iyer, N., Kalyanaraman, Y., Lou, K., Jayanti, S., and Ramani, K., (2003), "Early results with a 3D Engineering Shape Search System," International Symposium on Product Lifecycle Management (PLM '03), Indian Institute of Science, Bangalore, India.

16. [Lou04] Lou, K., Prabhakar, S., Ramani, (2004), "Content-based Three-Dimensional Engineering Shape Search," Proceedings of the 20th International Conference on Data Engineering (ICDE04), March 30 - April 2, 2004, Boston, USA, pp. 754-765.

A Multi-perspective approach for the design of Product Configuration Systems – An evaluation of industry applications

Lars Hvam

Centre for Product Modelling
Dept. of Manufacturing Engineering and Management
Technical University of Denmark
DK2800 Lyngby, Denmark
www.productmodels.org

This article presents a suggested procedure for modelling and implementation of product configuration systems. The procedure has been applied in several industrial companies. The procedure comprises:

- Analysis of the business processes to be supported by a product configuration system. Furthermore, a demarcation and definition of the configuration system to be designed.
- Analysis and modelling of the part of the company's product assortment which is to be included in the configuration system.
- Selection of configuration software and programming of the configuration system.
- Implementation of the configuration system.
- Operations, maintenance and further development of the configuration system.

The procedure has been developed on the basis of the general methods for modelling IT-systems (UML) as well as on the theory for modelling technical systems and theory for the development of business processes.

The procedure or certain parts of the procedure have currently been tested and further developed in cooperation with a number of industrial companies including F.L.Smidth, American Power Conversion (APC), Aalborg industries, NEG-Micon, GEA-Niro and IBM-SMS. This paper presents the experiences gained from 4 of these industrial companies which have applied the procedure. Based upon this experience the utility of the method is evaluated, and suggestions/ideas as to a further development of the method, e.g. as regards the integration of organisational and economical problems are collected.

Introduction

Customers world wide require personalised products. One way of obtaining this is to customise the products by use of product configuration systems (Tseng and Piller, 2003). Several companies have acknowledged the opportunity to apply IT-based configuration systems[2] to support the activities of the product configuration process. Companies like Dell Computer and American Power Conversion (APC) rely heavily on the performance of their configuration systems, as a configuration of their complex product portfolio would not be feasible if the product configuration processes should be carried out manually (Tiihonen et al., 1996).

A product configuration system is capable of supporting the activities of specifying products in sales, design and methods engineering – the specification process.

The activities in the "specification process" includes an analysis of the customer's needs, design and specification of a product variant which full-fill the customer's needs and specification of e.g. the products manufacturing, transportation, erection on site and service (specification of the product's life cycle properties). The activities in the specification process are characterised by having a relatively well-defined space of (maybe complex) solutions as a contrast to product development, which is a more creative process.

---

[2] E.g. Oracle Product Configurator, iBaan Configurator, and Siebel Configurator.

Typical goals for the specification process are the ability to find an optimal solution according to the customer's needs, high quality of the specifications, short lead time and a high productivity of the work carried out in the specification process.

This article focuses on the opportunities in supporting the specification process with IT. The idea is to formalise knowledge and information of the products and their life cycle properties and to express the knowledge in IT-systems – so-called product configuration systems.

A product configuration system is based on a model of the product portfolio. A product model can be defined as a model that describes a product's structure, function and other properties as well as a product's life cycle properties e.g. manufacturing, assembly, transportation and service (Krause, 1988, Mortensen, 2000, Hvam, 1999).

A product model used as basis for a product configuration system also includes a definition of the rules for generating variants in the product assortment (Mortensen, 2000, Schwarze, 1996).

A product configuration system is a knowledge integrated or intelligent product model, which means that the models contain knowledge and information about the products, and based on this is able to derive new specifications for product instances and their life cycle properties. The principle of using product configuration systems to support the specification process is to formalise the product knowledge, implement the product knowledge in an expert system and thus make the product knowledge of the engineers explicit to the rest of the organisation.

The use of product configuration systems is increasing. There are today a number of examples of product configuration systems which, for instance, are used to support sales, design of product variants and production preparation. Experiences from a considerable number of industrial companies have shown that often these product configuration systems are constructed without the use of a strict procedure or modeling techniques.

As a result of this, many of the systems are unstructured and undocumented and they are therefore difficult or impossible to maintain or develop further. Thus there is a need to develop a procedure and associated modeling techniques, which can ensure that the constructed product and product related models are properly structured and documented, so that the configuration systems can be maintained continually and developed further.

Another experience is that the product configuration systems are not always designed to fit the business processes, which they are meant to support. Finally, an important precondition for building product configuration systems is that the products are designed and structured in a way, which makes it possible to define a general master of the product, from which the customer specific products can be derived.

In order to cope with these challenges, a procedure for building product configuration systems should include: An analysis and redesign of the specification processes in focus, an analysis and eventually redesign/ restructuring of the product range to be modeled, and finally, a structured "language" - or modeling technique - which makes it possible to document the product configuration systems in a structured way.

This article presents a procedure for building product configuration systems. The procedure is based on several theoretical domains including:

- Modelling concepts – based on object oriented modeling
- Product analysis – dealing with the transformation of product knowledge into a product model.
- Organizational aspects – how to organize the development of product configuration systems, and the subsequent use of the systems.
- Development of business processes – how to identify and redesign new business processes supported by product configuration systems.

A PROCEDURE FOR BUILDING PRODUCT configuration systems

The procedure contains 7 phases. The starting point for the work is an analysis and redesign of the business processes, which will be affected by the product configuration systems (phase 1). In phase 2 the products are analyzed and described in a so-called product variant master. Phase 3 includes the final design of the product configuration system using the object oriented modeling techniques. Phases 4 to 7 deal with design, programming, implementation and maintenance of the product configuration systems. Phases 3 to 7 follow by and large the general object oriented project life cycle (Hvam, 1999) (Felfernig et al, 2000).

There may be some overlap and iterations between the individual phases. The procedure is shown in Figure 1.

Phase	Description
1	**Process Analysis** Analysis of the existing specification process (AS-IS), statement of the functional requirements to the process. Design of the future specification process (TO BE). Overall definition of the product configuration system to support the process. **Tools:** IDEF0, flow charts, Activity Chain, Model, key numbers, problem matrix, SWOT, list of functional describing characteristics and gap analysis.
2	**Product Analysis** Analysing products and eventually life cycle systems. Redesigning/ restructuring of products. Structuring and formalising knowledge about the products and related life cycle systems in a product variant master. **Tools:** List of features and product variant master.
3	**Object Oriented Analysis (OOA)** Creation of object classes and structures. Description of object classes on CRC-cards. Definition of user interface. Other requirements to the IT solution. **Tools:** Use cases, screen layouts, class diagrams and CRC-cards.
4	**Object Oriented Design** Selection of configuration software. Defining and further developing the OOA-model for the selected configuration software. Requirements specification for the programming including user interface, integration to other IT-systems.
5	**Programming** Programming the system based on the model. Testing the configuration system
6	**Implementation** Implementation of the product configuration system in the organisation. Traning users of the system, and further training of the people responsible for maintaining the product configuration system.
7	**Maintenance** Maintenance and further development of the product and product related models.

Figure 1. The procedure for building product configuration systems.

Process Analysis

Initially an AS-IS description of the current specification processes is made. This description should expose the structure between activities, people, IT systems, shifts of responsibility etc. Key figures for characterising quality, resource consumption and throughput times may support the description. Analysis tools to be used may be IDEF0 function modelling, The Activity Chain Model, or different kinds of flow charts.

The outset of the process analysis is a prioritised list of the critical goals of the specification processes in focus as well as an analysis of other future requirements sat by the surroundings. The present performance of the specification processes is measured and compared with the goals and other requirements through a so-called gap analysis. The gap analysis makes it possible to evaluate the gap between the current process performance and the required performance. This leads to identify the potential improvements to realize by using product configuration systems. To support the goal setting and the requirements analysis a list consisting of functional characteristics for the specification processes is used. The list of functional characteristics for the specification processes include among others:

- Kinds of input and output specifications
- Throughput time
- Resource consumption
- Quality of specifications
- Insight into consequences (simulation and optimisation of different product configurations)
- Flexibility of the process
- Frequency of similar specification activities

- Accessibility of knowledge

The functional characteristics are further outlined in (Hvam and Hansen, 1999).

Based on the process analysis a concept for a future ideal business process is designed. This ideal concept is made in order to be more creative and not so restricted by "historic" procedures in the company, - similar to the BPR approach (Hammer, 1990). With the ideal concept in mind a more realistic process design (TO-BE) is made. This TO-BE description will consist of structural elements such as:

- Sequence of activities in the future specification process (TO BE)
- Organisation of the specification processes
- The product configuration system (s) to support the specification process.

In relation to the definition of the future specification process, the product configuration system, which is to support the specification process are defined in overall by use of the framework for modelling configurable products presented in the next section and illustrated in figure 2 below.

Product analysis

In this phase the product range to be modeled is analyzed in order to gain an overview of the product families, and their life cycle properties. The analysis covers the product's function and structure, the properties of the products, the variations of the products (the generic level) and the related systems in the product life cycle. Figure 2 below shows a general architecture for describing a product range including the above mentioned views.

Figure 2. A framework for modelling a product range.

The framework in figure 2 is further outlined in (Hvam, 1994 and Hvam et al, 2004b).

Normally, product configuration systems only contain a minor part of the proposed views in the architecture. The specific views to include in the product model are defined based on the overall content of the product configuration system outlined in phase 1.

A formal way of describing the product assortment is a so-called product variant master (PVM). A product variant master consists of two main elements: a generic part-of structure and a generic kind-of structure.

Figure 3 below shows the principles of a product variant master.
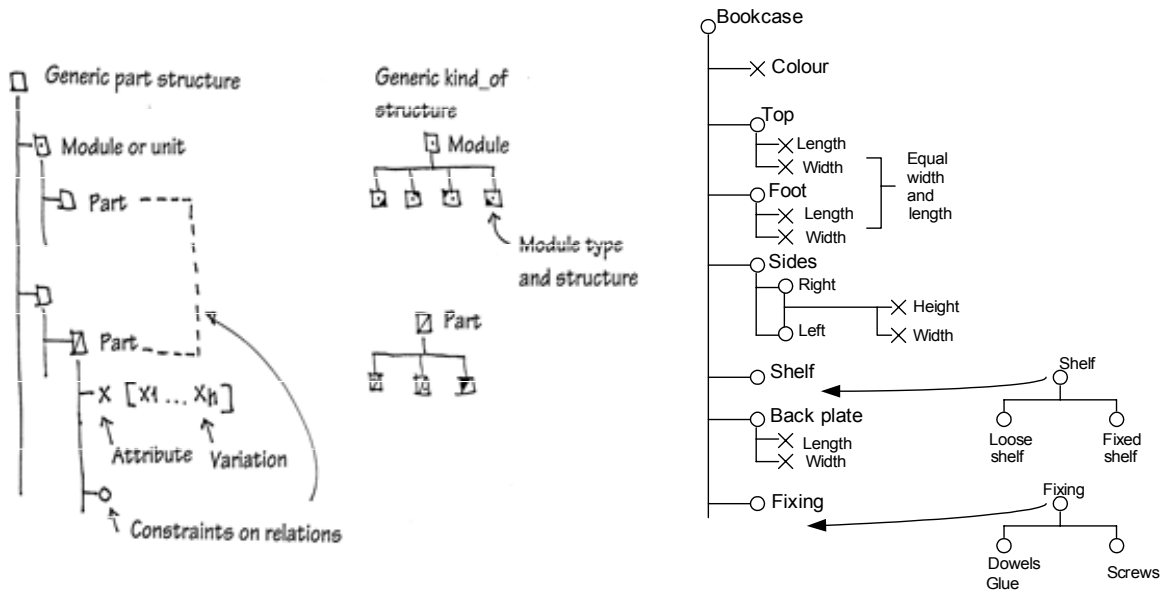
Figure 3.

The principles of a Product Variant Master (left). A PVM for a book case (right).

The left side of the product variant master (the part of structure) includes the modules or parts used in the entire product family. This is due to the aggregation structure in object oriented modeling. The right side of the product variant master (The kind of structure) includes the parts, which can be exchanged in the product. This is due to the specialization structure in object oriented modeling.

Experiences, from the 4 cases presented in chapter 3, showed that the product variant master was a very good tool for discussing the product assortment, i.e. where to start modeling the configuration system, which product variants shall be included, which product parts are stable and which product parts will change, etc. The product variant master is described by use of e.g. Visio™ or Excel™ and printed on a big piece of paper e.g. around 3 x 1 meters. The principles of the product variant master is further outlined in (Mortensen et al, 2000) and (Hvam et al, 2004b).

Object Oriented Analysis (OOA)

OOA is a method used for analysing a given problem domain and the field of application in which the IT system will be used (Bahrami, 1999), (Bennett et al, 1999), (Booch et al, 1999). The purpose of the OOA is to analyse the problem domain and the field of application in such a way that relevant knowledge can be modelled in an IT system. The problem domain is the part of reality outside the system that needs to be administrated, surveyed or controlled. The field of application is the organisation (person, department) that is going to use the system in order to administer, survey or control the problem domain.

Modelling the problem domain

The OOA model can be made through the activities described in figure 4, which describes the OOA as consisting of five phases or layers. These layers can be seen as different viewpoints, which together make up the OOA model. Normally the five activities are carried out through a top down approach, but there are no restrictions in that sense. Typically the OOA model will be the result of a number of iterations of the analysis process.
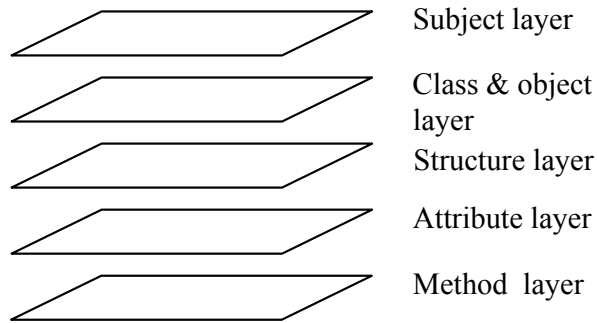
Figure 4. The five layers of OOA modeling (Coad and Yourdon, 1991)

The *subject layer* contains a sub-division of the model into different subject areas

The *class and object layer* contains a list of the classes and objects which have been identified in the individual subject areas.

The *structure layer* contains the relationships between the classes, i.e. a specification of generalisation and aggregation.

The *attribute layer* contains a specification of the information associated with the individual classes, i.e. what the classes know about themselves.

The *method layer* contains a description of the individual class' methods (procedures), i. e. what the class can perform.

Classes and structures are identified based on the product variant master from phase 2. The static structure is mirrored in the layers of theme, classes and objects, structure and attributes, while the more dynamic aspects in the model mainly are related to the method layer. The result of the OOA can be illustrated in a class diagram (Booch et al, 1999) and on a reworked version of CRC cards as shown in figure 5 below. A CRC card (Class-, Responsibility and Collaboration Cards) describes the details of the classes and contain a description of the mission of the class, the attributes and methods (constraints), the relations to other classes and often also a sketch of the product/ part in focus (Bellin et al., 1997; Hvam et al, 2003).

Figure 5. CRC card for product modeling.

The notation used in the class diagram is illustrated in figure 6, which shows a part of a class diagram for the bookcase based on the product variant master shown in figure 3. The notation is a part of the Unified Modelling Language (UML), which has been chosen since it is the preferred standard world wide and is used in many development tools.
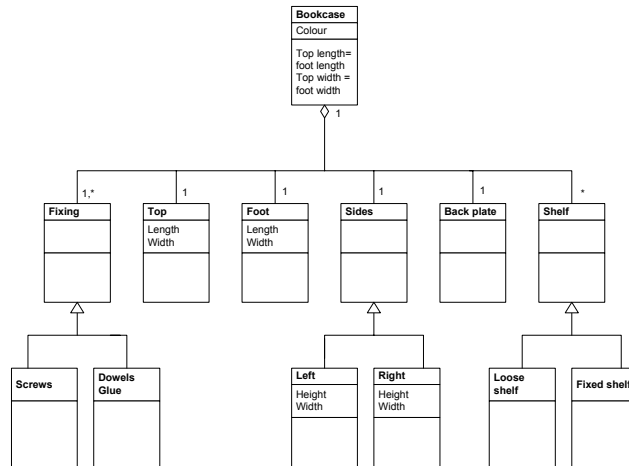
Figure 6. Class Diagram for a bookcase (UML)

Modelling the field of application

The second part of the OOA consists of an analysis of the field of application. Here the interaction between the user and the configuration system is analysed in order to determine the functionality of the system, the user interface, integration to other IT-systems etc. Other elements that need to be determined are also requirements to response time, flexibility and so on. The result of this is a description of the user interface and a requirement specification for the product configuration system.

Object Oriented Design (OOD)

Based on the OOA-model a standard configuration system is selected. A set of criteria's for selection of product configuration software has been developed (Hvam et al 2004b). In the design phase the perspective changes from being domain oriented (what and which task?) to implementation oriented (how?). According to (Coad & Yourdon, 1991) four perspectives are used during the development of the OOD model:

- *User interface*, which determines the user's communication with the system.
- *Problem domain*, in which the OOA model is corrected in accordance with design-specific criteria.
- *Data management*, where the structure of the stored data and methods for control of data is being modelled.
- *Task management*, which is used in cases where the system has to perform several tasks simultaneously (multitasking)

Object-oriented design contributes, like the other phases of the object-oriented project life cycle, to a structured procedure, and thus makes it possible to control the entire project more closely.

If a standard configuration tool is used (Baan Configurator, Oracle SellingPoint, etc.), most of the design parameters are frozen.

Programming

The product configuration system is programmed based on the models sat up through phases 2-4.

In the last few years a large amount of work has been done to create various standard configuration solutions. The major suppliers of ERP (Enterprise Ressource Planning) and Front Office systems are now joining the market. The list below illustrates the more famous actors in the market:

Front Office/ERP:	Front Office:
Baan (now Invensys CRM), Oracle, Cincom, Sap,	Siebel, Trilogy, Calico, Firepond, Selectia, BT Squared,

Intentia, i2 Technologies, J. D. Edwards, Peoplesoft etc.	Clarify, SalesLogix etc.

A standard configuration system makes the domain experts more independent of software programmers since the actual programming is relatively simple and can be done without extensive programming skills. However the integration of a standard system to other systems would normally necessitate the work from programmers.

Implementation

Implementation, user acceptance, maintenance and follow-up are very critical factors. The system must stand its trial here. User acceptance is completely crucial if the system is to be a success; if the users are not satisfied, the system will not survive long. One way of getting the users' acceptance of the system is to involve the users already in the analysis phase. This can be done by developing an early prototype of the system, which the users can comment on. Also training and current information of the users will facilitate the users' acceptance of the system.

Maintenance

Product configuration systems can be viewed as "living organisms." The models will soon loose their value if they are not further developed and maintained. The object oriented structure and documentation (the product variant master, class diagrams and CRC cards) of the product configuration system make it considerably easier to maintain and develop the configuration system further.

The application of product configuration introduces a new way of doing business. New tasks are introduced in the organisation. E.g. a salesman will have to use the product configuration system in order to configure a product, and a product designer (product manager) will have to build up and maintain the information and rules describing the products. This calls for commitment and ongoing motivation from the top management. Besides this both users and model managers need education and training in using and maintaining the product configuration systems.

Evaluation of the procedure based on industry applications

The industry applications

The procedure has been applied in several manufacturing companies. In the following I shall give a brief introduction to 4 of those companies, their configuration projects and discuss the experiences and lessons learned by using the suggested procedure for design and implementation of product configuration systems. The companies in focus are F.L. Smidth A/S, GEA Niro A/S, American Power Conversion A/S, and a small manufacturer of electronic switchboards.

Application of the procedure at F.L.Smidth A/S

As part of the Danish FLS Industries, F.L. Smidth is an engineering and industrial company with an international market leading position within the area of development and manufacturing of cement plants. F.L. Smidth has more than 100 years experience within the area, and after purchasing the American Fuller Company in 1990, the market share is now more than 50% with a turnover around 1 billion USD (2001).

A modern cement plant typically produces 2-10,000 tonnes of clinkers per day (TPD), and the sales price for a 4,000 TPD plant is approx. 100 million USD[3]. Every complete cement plant is customized to suit the local raw material and climatic conditions, and the lead-time from signing the contract to start-up is around 2½ years.

The process analysis focused on the quotation process. The quotation process is carried out in two steps. The first step is a so-called budget quotation including an overall dimensioning of the cement factory, a process diagram and a price estimate. The next step is a so-called detailed quotation including a detailed description of the processes and machines in the cement factory. The configuration project focused on the budget quotation because the budget quote included fewer details, and

---

[3] A 4,000 tonnes per day (TPD), complete kiln line, semi turn-key, service, supervision, vehicles, training, steel plates to local manufacturer, and civil design (1998 prices).

because all significant decisions as to the cement factory's capacity, emissions, total project costs etc. are made during the budget quotation. The process analysis revealed that the process of making budget quotations was very resource consuming, with a long lead time and leading to quotations of varying quality. A gap analysis indicated that the lead time for making budget quotations could be reduced from 3-5 weeks to 1-2 days, the resources spent could be reduced from 15-25 man-days to 1-2 man-days and finally by using a product configuration system it would be possible to optimise the cement factory with respect to e.g. capacity, emissions, price and the use of previously designed machines. The configuration project was initiated based on the process analysis and the setup of a prototype version of the product configuration system.

In the product analysis the cement factory was divided into 9 process areas. Based on the process areas the model was structured hierarchically starting with scope list and mass flow leading to selection of solution principles described as arrangements and lists of machines. The product analysis was carried out by using the product variant master and the CRC-cards. The product variant master proved to be a useful tool for modelling the overall cement milling processes and machine parts of the cement factory. The product variant master was built up through a series of meetings between the modelling team and the product experts at F.L.Smidth. The detailed attributes and constraints in the model were described on the CRC-cards by the individual product experts.

The product variant master and the CRC-cards formed the basis for the object oriented analysis. Based on the product variant master an object oriented class structure was derived and the CRC-cards were checked for inconsistency and logical errors. The product variant master is drawn by using Visio, the CRC-cards are typed in Word documents. The product configuration system was implemented in an object oriented standard configuration software "iBaan eConfiguration Enterprise." The OOA model was derived by the programmer of the configuration system. The product variant master and the CRC-cards still serve as the basis for communicating with the product specialists. The CRC-cards both refers to the product variant master and the object oriented class model, which represents the exact structure of the product configuration system in the "iBaan eConfiguration Enterprise."

The Implemention of the configuration system was divided into stages to allow for a smoother implementation. At first a beta version of the product configuration system was tested by representatives from the sales department. The test revealed the need for a rework of the structure in the product variant master and of the configuration system and a new version was tested and finally released for use. The product configuration system is currently being updated and further developed by a task force who is responsible for updating further developing the model and the configuration system in cooperation with the product specialists.

Lessons learned from applying the procedure at F.L.Smidth were that a clear demarcation of the product configuration system is crucial in order to secure that a product configuration project can be fulfilled in due time, and that the configuration system can be implemented in a standard product configuration system. The configuration system was defined by use of the framework for product configuration systems derived in (Hvam, 1994), by a definition of the product range to include in the product configuration system and by defining the level of details in defining the product assortment. The project has proven the use of the concepts; parts, functions and organs can be useful when modelling complex products. The project at F.L.Smidth is further outlined in (Hvam et al. 2004a). Finally the configuration system has been implemented in two different product configuration systems. The first configuration system was not object oriented, while the second configuration system ("iBaan eConfiguration Enterprise") was fully object oriented. Applying an object oriented system made it considerably easier to implement and maintain the product configuration system.

Application of the procedure at GEA Niro A/S

GEA Niro A/S is an international engineering company that has a market leading position within the area of design and supply of spray drying plants. World-wide the Niro Group of companies consists of 36 companies and 2050 employees creating approx. 340 mio. USD in turnover a year. The products are characterised as highly individualised for each project.

The configuration project at GEA Niro is in many ways similar to the configuration project at F.L.Smidth. The project focuses on the quotation process. The aim of introducing a product configuration system is to reduce lead times, and resources spent on making quotations, optimization of the spray drying plants and the formalisation of product knowledge, so it can be accessed by relevant persons in organisation. During the development of the product model the need for an effective documentation system has emerged. Early in the project it was decided to separate the documentation system from the configuration software due to the lack of documentation facilities in the standard configuration systems. Lotus Notes Release 5 is implemented throughout the company as a standard application, and all the involved people in the configuration project have the necessary skills to operate this application. The documentation tool is therefore based on the Lotus Notes application.

The documentation system is built as a hierarchical response system. The UI is divided into two main parts, the product variant master and the CRC cards. However the product variant master is used in a different way. Only the whole-part structure of the product variant master is applied in the documentation tool. Main documents and responses is attached to

the structure of the product variant master. The configuration system is implemented in Oracle Configurator. Since the standard configuration software does not provide full object orientation the CRC cards described in section xx has been changed to fit the partly object oriented system. The fields for generalization and aggregation have been erased. The aggregation relations can be seen from the product variant master and generalization is not supported.

To ease the domain expert's overview an extra field to describe rules has been added. In this way, methods (does) have been divided into two fields, does and rules. "Does" could be e.g. print BOM while "Rules" could be a table stating valid variants. The CRC Card is divided into three sections. The first section contains a unique class name, date of creation and a plain language text explanation of the responsibility of the card. The second section is a field for sketches. This is very useful when different engineers needs quick information about details. The sketch describes, in a precise manner, the attributes, which apply to the class. The last section contains three fields for knowledge insertion and collaborations. Various parameters such as height, width etc. which the class knows about itself are specified in the "knows" field. The "knows" field contains the attributes and the "rules" field describes how the constraints are working. The "does" field describes what the class does, e.g. print or generate BOM. Collaborations specify which classes collaborate with the information on the CRC card in order to perform a given action.

The documentation system has been in use throughout the project period. As mentioned, the documentation system was created by using standard Notes templates. This gives limitations according to functions of the application. Class diagrams are not included in the documentation tools. They must be drawn manually. Implementing class diagrams would demand a graphical interface, which is not present in the standard Lotus Notes application. The configuration system is to be implemented during the fall 2004. Since the configuration system is not fully implemented yet the maintenance routines have not been checked properly. The first feedback indicates a positive effect when developing the configuration system, and a more simple maintenance is forecasted even though the generation of a reminding email to document owners after defined intervals is yet to be tested. See (Hvam et al 2002), for further information of the configuration project and the documentation tool at GEA Niro.

Lesson learned from the project at GEA-Niro were that an IT-based documentation tool is necessary in order to secure an efficient handling of the documents in the product model (product, variant master and CRC-cards). The configuration system is implemented in Oracle Configurator, which is not a fully object oriented configuration system. This means that e.g. inheritance between object classes in the product configuration system is not possible. GEA Niro uses a variant of the product variant master, where only the whole part structure on the right side of the product variant master is applied. However the experiences from the project are that the revised product variant master and the CRC-cards still secure a structure and documentation of the system.

Application of the procedure at American Power Conversion A/S

American Power Conversion (APC) produces data centre infrastructure such as uninterruptible power supplies, battery racks, power distribution units, racks, cooling equipment, accessories etc. The total turnover is approx. 1,3 billion USD (2002). APC has applied the procedure for building product configuration systems since 2000. Today APC has 8-9 product configuration systems. APC has formed a configuration team with approx. 25 employees situated in Kolding, Denmark. The configuration team is responsible for development and maintenance of APC's product configuration systems, which are used worldwide.

The product configuration systems are an integrated part of APC's business setup. The products are sold through the product configuration system, which makes it possible for APC to control a huge amount of sales personnel around the world. The product configuration, including the work out of quotations and manufacturing specifications, is carried out by the configuration system saving considerably resources. The lead time for making quotations and manufacturing specifications is reduced from weeks to hours. And finally the product configuration systems make it easier to introduce new versions of the products to the sales personnel and the customers.

The product assortment is modeled in a co-operation between the configuration team and the product development teams. The product variant master and the CRC cards are used in the modeling process and document the configuration systems throughout programming and maintenance of the product configuration systems. Similar to GEA Niro, APC has developed a Lotus Notes based documentation tool – called the CRC-card database - to handle the documentation of the models. The configuration systems are implemented in Cincom Configurator. The configuration projects at APC are further described in (Hansen & Hvam, 2002).

Lessons learned from APC are, that the need for an IT-based documentation tool is even bigger at APC, than at GEA Niro. Running a configuration team with 25 employees, which have to communicate with product development teams around the world, requires a structured procedure for building the product configuration systems, as well as a Web-based documentation tool, which can be accessed by APC-employees worldwide. At APC a Notes based documentation tool has been developed similar to the Notes application at GEA-Niro. The documentation system at APC has now been running for 3-4 years. The experiences from running the documentation system is, that the structure in the product variant master and

the CRC-card database form a solid basis for communicating with e.g. product designers and for maintaining and further developing the product configuration systems. However, the fact that the documentation system is separated from the configuration software, means that the attributes and rules have to be represented in both the documentation tool and in the configuration with only limited possibilities of relating the two systems to one another. This means, that the configuration team at APC need to be disciplined about updating the documentation system every time a change is made in the configuration system.

Application of the procedure at a Danish manufacturer of electronic switchboards

The fourth case is a Danish company making electronic switchboards. It has more than 100 employees and a turnover of approx. 15 million Euros. The process analysis revealed that the lead-time for generating quotations was 3 to 5 days, and the company uses 2 to 4 man-hours for each quotation. The process leads to frequent errors, and often the time necessary for the optimization of the boards cannot be found.

By implementing a product configuration system the company gets a much more structured process flow where the company's knowledge regarding construction of an electronic switch board is made available to the customers, and complex calculations can be made very quickly. The desired effects for the company of the application of a product configuration system are identified as:

- A significant reduction of lead time for making quotations from 3-5 days to 10 minutes.
- A total elimination of resources spent for making quotations, as the customers now can configure an electronic switchboard on their own by using the product configuration software available on the company's homepage.
- The opportunity of optimizing the electronic switch boards with respect to e.g. heat loss and price.

The electronic switchboard consists among other things of an outlet field, a cable section, a measure field, a transformer field, an unmeasured power field, etc. The product configuration system must be able to calculate the number and dimensions of the distribution sections and the cable sections, the contents of the measure field, type and number of switches to be included in the outlet field, etc. During the product analysis a product variant master was created in cooperation with the sales people, product developers and production employees in the company. Details in the model were modelled by using the CRC-cards. Each CRC-card is attached to a node on the product variant master.

The final consists of a class diagram and the CRC-cards. The software from Invensys CRM / Baan was selected for the programming phase. There are approx. 12.000 boolic variables and 7.000 rules in the configurator. The configuration system has been implemented at Demex Electric during spring 2003.

Lessons learned from this project are that the product variant master and the CRC-cards formed a solid basis for involving relevant personnel in modelling the electronic switch boards. Application of product configuration systems was a new task for the company. The use of prototyping in the early phases of the project has proved to be a valuable means to introduce the concept of product configuration to persons with no or only little knowledge on product configuration. Furthermore prototyping was used to test how to implement details in the product model in the configuration software.

Conclusion

The proposed procedure is based on well known and proven theoretical elements; business process analysis, theory for structuring product knowledge, object oriented analysis, artificial intelligence and project management. The aim of the procedure is to serve as a guide for engineers working with design, implementation and maintenance of product configuration systems. The experiences from applying the procedure in the above mentioned 4 industrial companies show that the procedure contributes to define, structure and document the product configuration systems. The product variant master and the CRC-cards make it possible to communicate and document the product assortment including rules and constraints on how to design a customer tailored product.

The procedure is developed based on the unified process and the basic principles in object oriented modelling and programming. However only a very small part of the standard configuration systems based on rules and constraints and including an inference engine are fully object oriented. In order to meet this actual situation the product variant master and the CRC cards have been changed to fit into configuration systems, which are not fully object oriented. The experiences from applying the procedure in building configuration systems in non object oriented systems are positive. However structuring and maintaining configuration systems are considerably easier in an object oriented standard configuration system. Furthermore an integration of the documentation tool and the configuration systems would considerably ease the work of design, implementation and maintenance of product configuration systems.

Working with the configuration projects in the companies have revealed the need for a professional approach to handle the organisational changes related to the design, implementation, running and maintenance of product configuration systems.

The basic concepts in project management and change management have been used in the projects with positive results. However a scientific research task remains as to clarify the consequences of formalising product knowledge into IT-systems. Furthermore the projects have revealed the need for a set of guidelines on, how to facilitate the organisational change process towards formalising product knowledge into product configuration systems in an efficient way.

Another lesson learned from the projects is that the four companies all have a clear description of the impact in the overall business strategy of implementing product configuration systems. Implementation of a product configuration system is an important part of the business strategy. The goals of implementing a product configuration system have been clarified early in the development project. The top managers have demonstrated a clear commitment to the project and have allocated the necessary resources to the project.

The framework for structuring product configuration systems has been used both as a means to demarcate the configuration systems in phase 1, and to setup the structure of the configuration system in the phases 2 and 3. The experiences of applying the framework have been positive. However the framework needs to be described further and further tested.

References

1. Bahrami, A. (1999). Object Oriented Systems Development. Irwin/McGraw-Hill.

2. Bellin, D. & Simone, S. S. (1999). The CRC-card Book. Addison-Wesley.

3. Bennett, S., McRobb, S. & Farmer, R. (1999). Object-Oriented Systems Analysis and Design using UML. McGraw-Hill.

4. Booch, G., Rumbaugh, J. & Jacobson, I. (1999). The Unified Modeling Language User Guide. Addison-Wesley.

5. Cawsey, A. (1998). The Essence of Artificial Intelligence. Prentice Hall.

6. Coad P. & Yourdon E. (1991); *Object-oriented analysis*; prentice-hall, Inc., second edition, 1991.

7. Felfernig, A., Friedrich, G. E. & Jannach, D. (2000). UML as Domain Specific Language for the Construction of Knowledge-based Configuration Systems. In International Journal of Software Engineering and Knowledge Engineering, volume 10/4.

8. Gilmore, J. & Pine J. (2000); *Markets of One, - Creating Customer Unique Value though Mass Customization*, Harvard Business Review, Boston, 2000.

9. Hammer, M. (1990): Re-engineering work: don't automate, obliterate, Harvard Business Review, July-August, 1990.

10. Hansen, B., Hvam, L. (2002). Design of Configuration Systems at APC. Proceedings of 15th European Conference on Artificial Intelligence. Lyon, France.

11. Hvam, L. (1994). Application of product modelling – seen from a work preparation viewpoint. Ph.D. Thesis. Dept. of Industrial Management and Engineering, Technical University of Denmark.

12. Hvam L. & Hansen B.L.(1999); Strategic guidelines for application of product models; The 4th Annual International Conference on Industrial Engineering Theory, Applications and Practice, San Antonio, Texas, November 17-20 1999.

13. Hvam, L. (1999). A procedure for building product models. Robotics and Computer-Integrated Manufacturing, 15: 77-87.

14. Hvam, L. & Malis, M. (2002). A Knowledge Based Documentation Tool for Configuration Projects. In Mass Customization for Competitive Advantage (Ed.: M.M. Tseng and F.T. Piller).

15. Hvam, L. & Riis, J. (2003). CRC-card for product modeling. Computers in Industry, volume 50/1.

16. Hvam, L., Malis M., Hansen B.L. & Riis J.(2004a); Reengineering of the quotation process - Application of knowledge based systems; Business Process Management Journal, Vol 10 issue 2, April 2004, pp. 200-213. ISSN 1463-7154

17. Hvam L., Mortensen N.H. and Riis J.(2004b); Product configuration; Centre for Product Modelling, Department of Manufacturing Engineering and Management; Technical University of Denmark, January 2004, 329 pages (In Danish).

18. Jackson, P. (1999). Introduction to Expert Systems. 3rd edition. Addison-Wesley.

19. Krause, F.L., Kimura, F. & Kjellberg, T. (1993). Product Modelling. In Annals of the CIRP, volume 42/2.

20. Mortensen, N.H., Yu, B., Skovgaard, H.J. & Harlou, U. (2000). Conceptual Modeling of Product Families in Configuration Projects. Proceedings of Product Models 2000-SIG PM. Linköping, Sweden.

21. Schwarze, S. (1996). Configuration of Multiple-Variant Products. BWI, Zürich.

22. Tiihonen, J., Soininen, T., Männistö, T. & Sulonen, R. (1996). State-of-the-practice in Product Configuration – a Survey of 10 Cases in the Finnish Industry. In Knowledge Intensive CAD, volume 1 (Ed.: Tomiyama, T., Mäntylä, M. & Finger S.). Chapman & Hall, pp. 95-114.

23. Tseng Mitchell M. and Piller, Frank. T. eds(2003); The Customer Centric Enterprise – Advances in Mass Customization and Personalization; Springer Verlag. ISBN 3-540-02492-1.

24. Whitten, J.L, Bentley, L.D. & Dittman, K.C. (2001). Systems Analysis and Design Methods. Irwin/McGraw-Hill, New York, USA.

A MULTI-AGENT BASED CONFIGURATION PROCESS
FOR MASS CUSTOMIZATION

Thorsten Blecker*, Nizar Abdelkafi*, and Gerold Kreutler**

*Department of Production/Operations Management
** Department of Computer Science and Manufacturing
University of Klagenfurt
Universitaetsstrasse 65-67
9020 Klagenfurt, Austria
Email: blecker@ieee.org,{nizar.abdelkafi, gerold.kreutler}@uni-klu.ac.at

Large product variety in mass customization involves a high internal complexity level inside a company's operations, as well as a high external complexity level from a customer's perspective. In order to reach a competitive advantage through mass customization, it is necessary to cope with both problems. This is done within the scope of variety formation and variety steering tasks: Variety formation supports customers during the configuration task according to their preferences and knowledge, variety steering tasks internally deal with finding the customizer's optimal offer. Driven by this economic background, we present a comprehensive multi-agent based design for a configuration process in this paper. It is identified as a suitable solution approach integrating both perspectives. The mass customized products are assumed to be based on a modular architecture and each module variant is associated with an autonomous rational agent. Agents must compete with each other in order to join product variants which suit real customers' requirements. The negotiation process is based on a market mechanism supported by the target costing concept and a Dutch auction.

Significance: The proposed multi-agent based configuration process enables the achievement of mass customization by reducing the external complexity from customer's perspective and supporting managers in variety steering decisions which aim at reducing the internal complexity from a company's perspective.

Keywords: Multi-agent systems, Configuration process, Market mechanism, Mass Customization

1. INTRODUCTION

Mass customization is a business strategy that aims at satisfying individual customer needs nearly with mass production efficiency (Pine, 1993). The development of mass customization is mainly due to the advances realized in modular product architectures and flexible manufacturing systems. However, the progress in the fields of information technologies and artificial intelligence for the support of the interaction process between the supplier and customer can be considered as the most relevant enabler for a successful implementation of the strategy.

In opposition to mass production, mass customization aims at providing customer-focused products with a high individuality level. Consequently, mass customization involves a variant-rich production environment, in which each produced variant can be a unicum. But customers generally do not seek out variety per se. They do not want more choices; they only want the choice that fits to their needs (Pine and Gilmore, 1999). That is why the main goal of mass customization should be „…that nearly everyone finds exactly what they want" (Pine, 1993, p. 44).

The resulting variety in mass customization triggers a high internal complexity level that leads to high overhead costs. Furthermore, excessive variety generally confuses customers who are overwhelmed by the complexity of the decision making process. This is basically due to the limited human information processing capacity and lack of technical product knowledge. Thus, the main objective should be to find an optimal product variety, which takes the company's and customer's perspectives.

From this point of view, we can identify two challenges. First, the mass customizer should be supported by an information system to efficiently cope with variety. Second, it is relevant to assist customers with adequate information tools in order to lead them in a fast-paced manner and with a low amount of effort to their optimal choice. On the basis of a multi-agent based configuration process, we provide a comprehensive system that copes with both identified challenges of internal and external complexities.

2. MAIN ASSUMPTIONS AND DESCRIPTION OF THE BASIC IDEA

Common configuration systems for mass customization necessitate product specific knowledge and often overstrain customers (Blecker et al., 2004). These systems should be improved to better support customers during the configuration process. For this reason, we opt for systems that are capable of assisting customers through advisory by capturing a customer's preferences and profile in order to display only the subset of the relevant product variants. From the mass customizer's product assortment, the best product variants succeed to be displayed to customers. Consequently, in the long term these product variants will better contribute to a supplier's success, which would suggest the retention of these variants in the mass customizer's offer. Those which are not short-listed rather trigger high complexity and are not relevant for customers. These product variants should be eliminated because they are neither beneficial for the supplier nor for the customer. If we suppose that each product variant strives for ensuring its existence in the mass customizer's offer for a long period of time (self-preservation principle), this can be interpreted as if the product variants would compete with each other in order to survive. Thus, it is relevant to define a mechanism defining the rules according to which the competition between variants is organized. The underlying idea is to consider a market mechanism supported by multi-agent technology.

It is assumed that the product assortment (i.e. solution space) of the mass customizer is based on a modular design, which means that the different product variants can be built by mixing and matching different module variants. This assumption is legitimate because the best method to achieve mass customization is to develop products around modular architectures as stated by Pine (1993). By taking this relevant assumption into account, it is more advantageous to assign an autonomous rational agent to a module variant rather than to a product variant. In effect, the number of product variants is commonly very high in mass customization and can go to billions as e.g. in the automobile industry. By considering each product variant as an autonomous rational agent, the configuration problem can explode and no sound solutions can be determined. However, by associating autonomous agents with module variants, the number of agents (called module agents) in the multi-agent population can be kept at a reasonable level, so that good solutions can be generated with acceptable computation times. Thus, the competition will occur between the module variants that will strive to form product variants that are displayed to customers.

A further important assumption for the definition of the multi-agent system is that we suppose the existence of common modules on which basis the product families are built. In business administration, these common modules are called platforms. Therefore, we call the corresponding agents platform agents to make the distinction vis-à-vis the module agents. The platform agents differ from the other agents, in that their life cycles are much more longer. Indeed, the development of a product platform is cost-intensive and necessitates many years of work. It is designed to represent the basic module of a product family for a long period of time. Consequently, the platform agents should not compete with each other to ensure their self-preservation. Due to this important property, we argue the high relevance of the platform agents during the negotiation process that is required for reaching agreements between the module agents.

The multi-agent based system should dynamically support each user during the online configuration process. This means that the system should iteratively generate and refine product variants according to specific customer needs. We call variety formation the process, in which the agents cooperate with each other in order to form customer-focused product variants. Furthermore, we will intend to design the agents so that they provide information about itself that helps managers evaluate the suitability of the module variant in the general fulfilment of customer requirements. This can have an important operational consequence that is the elimination of the module variants corresponding to unsuccessful module agents. In this context, we should note that the decisions that relate to the elimination or retention of product or module variants are made within the scope of variety steering.

3. INFORMATION SYSTEM FRAMEWORK FOR A MULTI-AGENT BASED CONFIGURATION

The module and platform agents are relevant elements in the market mechanism. However, to ensure that this market efficiently works, a suitable support has to be provided by an appropriate information system framework (Figure 1). In addition to the module and platform agents, this framework contains four other agents, namely the target costing, auction, product constraints' and validation agents, as well as two other components, namely the advisory component and configurator.

The advisory component is the starting point of the elaborated framework. Blecker et al. (2004, p. 4) define advisory systems for mass customization as "…software systems that guide customers according to their profile and requirements through a "customized" advisory process ending with the generation of product variants which better fulfill the real customer needs. In addition, they are customer oriented and do not assume any specific technical knowledge of the product." The advisory component ensures the online communication by initiating an advisory dialog, in which customers are asked targeted questions. Based on the customers' answers, the advisory component accordingly adapts the dialog in order to better elicit customer requirements. The advisory component maps a customer's requirements onto product functionalities (Blecker et al., 2004). That is why a software agent is needed to map functional requirements onto a technical product description. In this context, target costing (Seidenschwarz, 2001) provides many interesting insights.

Target costing is basically defined as a cost management tool for reducing product costs over the entire product life cycle. It also provides a calculation method for improving product design and planning. Based on the target selling prices that the customer would accept and the target profits that the supplier would like to achieve, the target costs are estimated. To apply the method, it is necessary to determine the product sub-functions and to appreciate their contribution in percent to the overall product function. Furthermore, each product component has to be evaluated with respect to its contribution level to the fulfillment of each sub-function. Thus, starting from the product target cost, it is possible to estimate the allowed costs of each product component. By considering target costing as a process, two main results are attained, namely a target cost of each component, as well as a procedure that maps product functions onto technical product specifications.

By adequately adapting the target costing concept to our specific case, we provide a method that can automatically and dynamically translate the functional characteristics to product-oriented specifications. The software agent that takes over the execution of this method is referred to as the target costing agent. It receives the product sub-functions from the advisory component and assigns to each of them the best module variants by taking into account the price range the customer would be willing to pay. However, the specification of the technical requirements at the product level would not result in module variants that exactly form one unique end product. In effect, due to the modular architecture, several product variants can be generated. Therefore, a module variants' refinement process should take place. This is ensured by the auction mechanism and validation agent that will be described later on. After translating the functional description to technical product specifications and determining the allowed costs, the target costing agent has to carry out three main tasks that consist of:

- Selection of the main platform agents that would best contribute to the fulfillment of a customer's profile. The selected product platforms can be chosen more than once, so that many product variants can be formed on the basis of one platform,
- Communication of the module agents that would be suitable to fulfill a customer's requirements to each selected platform agent.
- Communication of the module variants' cost ranges to the auction agent. On the basis of these costs the auction functions are determined.
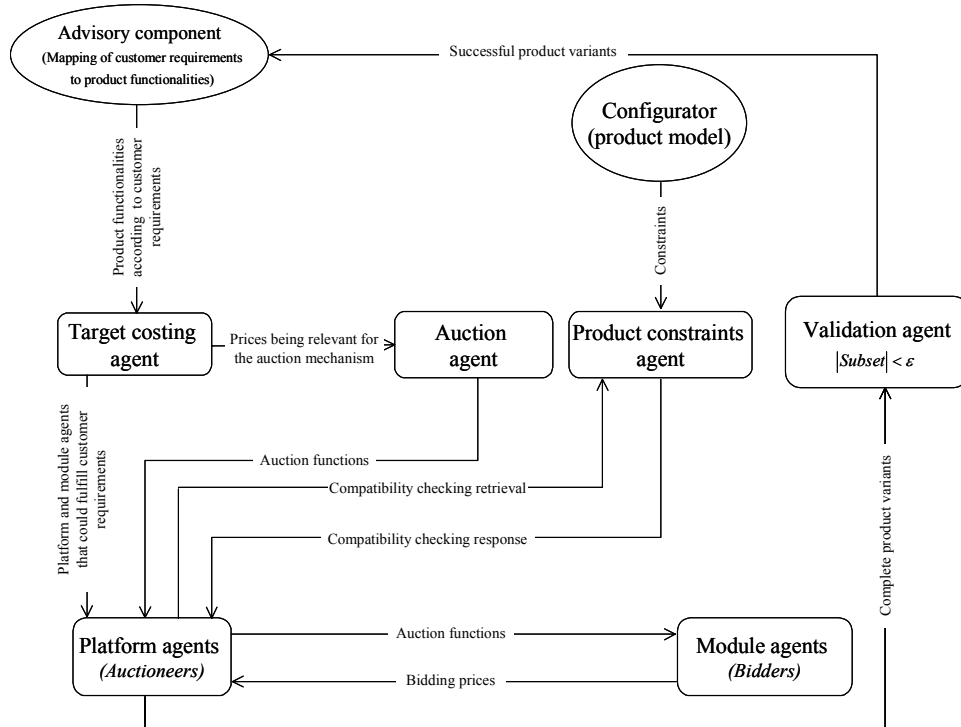
Figure 5: Framework supporting the multi-agent based configuration

In the outlined framework, the auction agent plays a relevant role. It receives the module variants' costs from the target costing agent and derives the auction functions, on which basis the platform agents initiate the different auctions. Note that the auction functions specify the protocol according to which the auctions run. According to their strategies, the module agents bid in order to participate at the formation of product variants. The platform agents as auctioneers also regulate the bidding process. When a module agent wins the bid, the platform agent asks the product constraints' agent to verify if the corresponding module variant violates the constraints. If there are no compatibility problems between the module variants,

the module agent is allowed to join. Otherwise, the module agent is rejected and the auction will continue normally until a module agent representing a module variant with no compatibility problems wins the bid. Thus, to ensure that only consistent product variants form, the product constraints' agent must have direct access to the configurator's logic.

After all of the bidding processes terminate, it may result that some product variants are incomplete because some relevant modules are missing. These are excluded and only the complete product variants will be communicated to the validation agent. This agent selects the product variants' *subset* that consists of the end products with the best specifications to suit customers' requirements. The advisory component takes over the task of displaying to customers the narrow range of selected product variants. Subsequently, the external complexity is reduced and customers can make a better buying decision. In the following, the most important agents for defining the market mechanism, namely the module agents, platform agents and auction agent are described with respect to their main capabilities, tasks and properties.

3.1 Module Agents

The module agents strive for survival and self-preservation within the variety offer of the mass customizer. In other words, they have the desire to ensure their existence for a long period of time. With respect to the self-preservation principle, as well as in order to regulate the bidding mechanism, we introduce the notion of a module agent's account, in which a fictive money amount is stored. By introducing a new module variant, a starting sum of monetary units is allocated to the agent. The agent's account is supposed to linearly decrease in the course of time. When the module agent's account is equal to zero, it risks "death". Within the scope of variety steering decisions, this means that managers have to consider the module variant for eventual elimination.

To ensure their survival, module agents must have the possibility to compensate their continuously decreasing accounts. Therefore, a reward mechanism has to be established. However, by winning a bid, the module agent pays the auction price that can be considered as a fee of participation. At the end of all auctions, several product variants form. The number of product variants that come through the validation agent is inferior or equal to the number of all formed variants. Thus, the collected sum of monetary units from all bidding agents will be attributed to the few remaining variants. Subsequently, the amounts are divided on the module agents of the successful product variants. The reward that the module agent receives is equal to the difference between what it received and what it originally paid as a fee of participation. That is why some agents draw a profit, whereas some others incur a loss. The account of the unsuccessful module agents diminishes more rapidly than those with positive rewards. Note that the life of the module agent that makes more profits in the course of time is longer than the life of the module agent that incurs a loss.

In the previously described framework, a module agent is allowed to simultaneously bid in many auctions. According to the principle of rationality, each agent wants to maximize its utility. Thus, the module agent must be able to not only appreciate the bidding behavior of other agents, but also to estimate as to whether a product variant would come through the validation agent or not.

3.2 Platform Agents

They are the auctioneers who initiate many online auctions on the basis of the bidding functions that are received from the auction agent. Platform agents only handle product specific information, but no customer requirements. As aforementioned, during auctions the platform agents communicate with the product constraints' agent in order to only ensure the participation of compatible module agents. Consequently, only consistent product variants result from the auctioning process. In contrast to module agents, platform agents do not have to strive for their survival, because the development of platforms is cost-intensive and long-planned. Therefore, it can be assumed that platform agents dispose of an infinite account of monetary units.

3.3 Auction Agent

The most common auctions being successfully implemented in real world transactions are: the English auction, Dutch auction, first-price sealed-bid auction and Vickrey auction (Klemperer, 1999). In order to determine the auction mechanism that is suitable for the described case, it is relevant to identify the main characteristics to be fulfilled by the auction:

- The auction mechanism should enable a progressive formation of product variants. This enables platform agents to check compatibility while product variants form.
- The auction should enable one to ascertain in advance the auction length so that it is possible to adjust auctions in such a way that they terminate at the same time. It is relevant that product variants simultaneously form because the successful ones should be displayed at once to the customer.
- The auction should enable module agents to track the product variants while forming. In this way, module agents are able to better evaluate their chances of success.

- The auction should drive the module agents to bid as early as possible. Because of the product constraints, a module variant that bids early tends to better ensure its participation and avoids the restrictions that could be imposed by other module agents.

With respect to the first criterion, first-price sealed bid and Vickrey auction are not suitable auction mechanisms. These cannot enable product variants to form progressively because the information of bidders is not open. Furthermore, the disadvantage of the English auction is that it does not permit one to estimate in advance how long the auctions will last. However, the Dutch auction mechanism fulfills all of the proposed requirements. It is an open auction where product variants can progressively form. The duration can be well adjusted by fixing the starting price and the decreasing money amount in the course of time. Moreover, agents should bid as early as possible in order to increase their chances in winning a bid. The corresponding auction agent in the proposed framework is called a Dutch auction agent. It defines the functions with which the platform agent initiates the bidding process. Platform agents communicate these functions to all module agents which are allowed to join the auction.

The Dutch auction functions should have the following properties: (a) the start value is derived from the value of the product function provided by the target costing agent, (b) the initial price must be harmonized with the module agents' starting accounts, and (c) all auctions of the same customer session must end at approximately the same time. (a) implicates that more valuable product functions are more expensive for module agents. This is legitimate because the possible rewards are consequently higher. (b) demands that the Dutch auction agent should have the capability of mapping the monetary value provided by the target costing agent to the units used in the auction process. Therefore, the Dutch auction agent also plays an important interfacing role between all agents and the advisory component (i.e. the user). (c) is required because the auction process carries out product variant formation in real-time. The customer concurrently obtains product suggestions. This demands that all auctions terminate at the same time to avoid delays.

4. DESCRIPTION OF THE MODULE AGENTS' BEHAVIOR FOR REAL-TIME CONFIGURATION

The bidding strategy of the module agent is formulated on the basis of the agent's own account and the bidding behavior of other agents in the environment. The account is private information of the module agent as usual in real-world auctions, which means that each agent does not have complete information. However, a module agent has the capability of tracking its environment in order to anticipate the behavior and the actions of opponent agents. For each module agent, we define two types of strategies, which are the long-term and short-term strategy.

The long-term strategy refers to the plan that the module agent sets in order to achieve its fundamental objective of ensuring a lengthy existence. To define the long-term strategy, variety managers have to ascertain two main entries for a module agent, namely $Acc\ (t=0)$ and T_∞. $Acc(t=0)$ is the starting account that the module agent receives at the beginning of its life cycle. It determines the period of time T that the agent would survive even when it participates in no auctions. However, T_∞ represents the period of time that the agent strives to survive. Within the scope of variety steering decisions, T_∞ is a relevant parameter that should be appropriately and carefully ascertained by variety managers. Owing to the rationality principal, the module agent strives to gain a sum of money called *Profits* that will ensure its survival until T_∞. The term $\dfrac{\Pr ofits}{T_\infty}$ indicates how much the module agent has to win per unit of time in order to achieve T_∞. By assuming that the risk of failing is equally distributed among all agents, the agent with higher $\dfrac{\Pr ofits}{T_\infty}$ will tend to participate at more auctions and has to be more aggressive.

The aggressiveness level (Benameur et al., 2002) of a module agent characterizes its intention to bid rapidly in order to participate in product variants. We argue that the richer a module agent is, the more it tends to be aggressive. Thus, the aggressiveness level of a module agent depends both on its account in the course of time, as well as the sum of money it has still to gain in order to achieve T_∞. To connect the long-term and short term strategies of a module agent, we introduce the notion of a budget. A budget is the sum of money that the agent allocates to bidding in the currently running auctions. The allocation of budgets depends on the survival strategy of the module agent who tries to allocate budgets in such a way that it can reach T_∞.

Suppose that there are *n* auctions that begin at the same time *t=0,* in which the module agent would like to participate. According to its strategy and aggressiveness level, the module agent ascertains a budget for these auctions. At the beginning of the auctions, the module agent sets a plan and ranks the bidding times in an ascending way. When the auctions run, the module agent may lose the first bid it has planned, or it is no longer allowed to participate because of product constraints. Then, the module agent can reallocate the available budget differently for the participation in the remaining auctions. This process continues until all product variants are formed.

5. THE OVERALL CONFIGURATION PROCESS

An overview of the multi-agent based configuration process is provided by Figure 2. This process is described by four main steps. (1) The advisory component, the target costing agent and the Dutch auction agent prepare relevant information for carrying out auctions. Note that the information issue from the target costing agent does not exactly determine one single product variant, but several ones may be possible. The target costing agent can specify a specific set of module agents or even one particular agent. This essentially depends on the sharpness of information that is gained from customers. (2) Then the multi-agent system based on the auction market mechanism takes over the task of determining consistent and complete product variants that could fulfill customers' requirements. (3) These product variants are submitted to the validation agent who (4) selects the best ones to be displayed to customers by means of the advisory component.
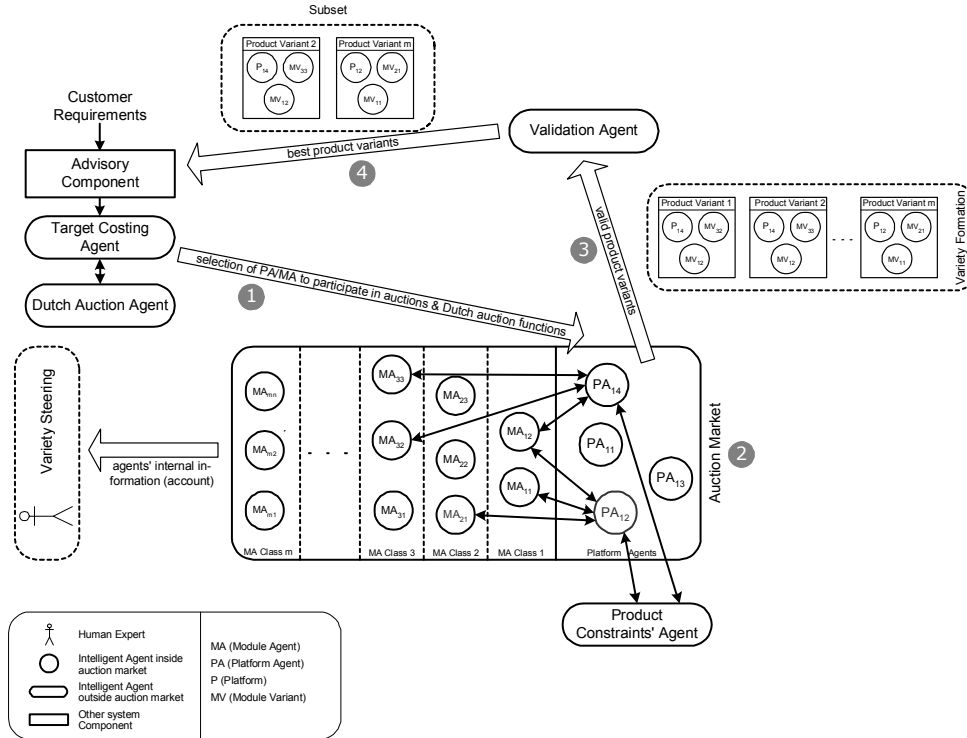
Figure 2: The multi-agent based configuration process

It is important to recall that the account of an agent is of high relevance within the scope of variety steering. Furthermore, the elimination of one single module variant will automatically trigger the elimination of several product variants. Consequently, internal complexity which is experienced inside a company's operations and manufacturing-related tasks is strongly reduced.

6. BENEFITS OF THE MULTI-AGENT SYSTEM IN VARIETY STEERING

The multi-agent based configuration process has a further advantage, in that each module agent provides information about itself that can be used by the variety manager in variety steering. Variety steering depends on two values that must be carefully ascertained by variety managers, namely $Acc(t=0)$ and T_∞. Both values strongly consider decisions in business administration, which are especially relevant to organize the multi-agent system. $Acc(t=0)$ determines how long a module variant is allowed to exist in the supplier's offer even when it participates in no product variant. On the other hand, T_∞ is the longest period of time, during which managers estimate that the module variant will be able to contribute to the fulfillment of customer requirements. Because of e.g. new technological innovations or changes in customers' preferences it is expected that after this period the corresponding module variants should be excluded from the offer. It is worth noting that when the account reaches the value of zero, the module variant should not be automatically eliminated. Human managers should carefully examine this decision.

From a business administration point of view, this concept is a novelty. In the technical literature, it is more often mentioned that the product variants that are not purchased by customers should be selected for eventual elimination (e.g. Wildemann 2000). However, this method is disadvantageous because the single truth that one can draw from the elimination of a product variant is that this variant no longer exists in the assortment and its turnover drops out (Lingnau 1994). The introduction of the multi-agent based configuration enables one to avoid the outlined shortcomings. The variety steering decisions are no longer based on the determination of which variants are purchased or not, but mainly on the variants reaching the final subset. This is more advantageous because the module agents participating in the formation of product variants that could be optimal from a customers' perspective have the chance to exist longer.

7. CONCLUSIONS

In the web-enabled mass customization two main challenges referred to as internal and external complexity are identified. External complexity is faced by customers and is essentially due to the huge number of product variants. It makes customers unable to meet optimal buying decisions. In contrast, internal complexity is experienced inside operations and manufacturing-related tasks. The external complexity problem is solved within the scope of variety formation, whereas the internal complexity is dealt with by variety steering.

It was assumed that products are built on the basis of a modular and a platform strategy. Furthermore, autonomous rational agents are associated with each module variant and platform. On this basis, the main framework describing the information system was presented. Besides the module and platform agents, the framework contains a configurator containing the product model, advisory system, target costing agent, auction agent, product constraints' agent and validation agent. We have shown how all of these components and agents communicate and interact with each other.

The main software systems in the proposed framework are the auction, the module and the platform agents who play a very important role in the multi-agent based configuration process. All of these agents are described with respect to their main properties, characteristics and actions. The main idea is that the module agents bid to take part in product variants that best suit customer requirements. The different auctions are initiated by the platform agents who receive the auction functions from the auction agent. Furthermore, it was identified that the Dutch auction is the most appropriate auction mechanism for coordinating the described multi-agent system.

By elaborating the multi-agent system, the main concern was to provide a system that helps optimally achieve the web-enabled mass customization. Because of the complexity of the mentioned problem, the developed system is also complex. However, the theoretical system design is robust and feasible. The difficulties that can be met during implementation basically concern the determination of the initial values of the different parameters which are defined to make the system work. Therefore, in order to cope with this problem, simulations will be necessary for setting up these values.

8. REFERENCES

1. Benameur, H., Chaib-draa, B. and Kropf, P. (2002): Multi-item auctions for automatic negotiation. Information and Software Technology, 44: 291-301.

2. Blecker, Th., Abdelkafi, N., Kreutler, G. and Friedrich, G. (2004). An Advisory System for Customers' Objective Needs Elicitation in Mass Customization. Proceedings of the 4th Workshop on Information Systems for Mass Customization (ISMC 2004) at the fourth International ICSC Symposium on Engineering of Intelligent Systems (EIS 2004), University of Madeira, Funchal/Portugal.

3. Klemperer, P. (1999). Auction Theory: A Guide to the Literature. Journal of Economic Surveys, 13: 227-286.

4. Lingnau, V. (1994). Variantenmanagement: Produktionsplanung im Rahmen einer Produktdifferenzierungsstrategie. Erich Schmidt Verlag GmbH & Co, Berlin, Germany.

5. Pine II, J. (1993). Mass Customization: The New Frontier in Business Competition. Harvard Business School Press, Boston, USA.

6. Pine II, J. and Gilmore, J. (1999). The Experience Economy. Harvard Business School Press, Boston, USA.

7. Seidenschwarz, W. (2001). Target Costing. Marktorientiertes Zielkostenmanagement. Vahlen, Muenchen, Germany, 2nd edition.

8. Wildemann, H. (2000). Komplexitaetsmanagement: Vertrieb, Produkte, Beschaffung, F&E, Produktion und Administration. TCW Transfer-Centrum, Muenchen, Germany.

Data Model and Personalized Configuration Systems for MASS CUSTOMIZATION – a Two Step Approach for Integrating Technical and Organisational Issues

Wolfgang Hümmer, Christian Meiler, Andreas Dietrich, and Sascha Müller

Department of Database Systems (Inf 6)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstraße 3
91058 Erlangen, Germany
Email: {huemmer, christian.meiler, sascha.mueller}@informatik.uni-erlangen.de

Lehrstuhl für Wirtschaftsinformatik II
Universität Hohenheim
Schwerzstraße 35
70599 Stuttgart, Germany
Email: Andreas.Dietrich@uni-hohenheim.de

Mass Customization (MC) is a management concept which aims at the combination of mass production and customer related individualization. Product configuration is mentioned as one of the critical factors in realizing MC projects but it is a summarizing term with different requirements and functions depending on the particular MC scenario. In this paper, an approach is presented integrating the technical and organizational aspects of product configuration for MC. Here, two basic aspects are important: customer cooperation in establishing product models and afterwards supporting customers by personalized configuration processes. For this purpose, we introduce a meta model that allows to specify highly customizable product families restricted by complex dependencies called implications. The novelty of our approach is that the customers are enabled to model the space of products fitting their needs using the same model and method as the supplier side. Based on these models a personalized configurator is automatically generated.

Significance: Product configuration for Mass Customization means instantiating a particular product specification from a defined solution space. Therefore, this paper focuses on building product structures for mass-customizable products (static view) and on a matching the demand of the customers against the supply of enterprises (dynamic view).

Keywords: product modeling, meta model, mass customization, personalized configuration

1. INTRODUCTION

Mass Customization (MC) is a management concept which aims at the combination of offering highly personalized products and realizing manufacturing efficiency as known from mass production (e.g. low production costs). However, each order is treated separately for achieving individuality within the product and reaching an optimal fit between the characteristics of the product and the demander's needs. Product configuration is mentioned as one of the critical factors in realizing MC projects [Piller, F. and Stotko, Ch. M. (2002)].

From a customer point of view, the limiting factors for high satisfaction are predefined individualization parameters and their values. Creating optimal product structures is one of the major challenges for enterprises. Up to now, there has been a lot of work on technical aspects of product configuration: formal methods, implementation concepts, usability, virtual product modeling etc. In our paper, we propose an approach integrating the technical and organizational aspects of product configuration for MC. We focus on two basic aspects: customer cooperation in establishing product models and afterwards supporting customers by personalized configuration processes.

For this purpose, we introduce a meta model that allows to specify highly customizable product families restricted by complex dependencies [Dietrich, A. and Hümmer, W. and Meiler, C. (2004)]. The novelty of our approach is that the customers are enabled to model the space of products fitting their needs using the same model and method as the supplier side. Therefore, it is possible to automatically match demand and supply by means of evolutionary algorithms. From the remaining open issues, i.e. those issues that could not be automatically resolved by the match making process, our approach allows for automatically generating personalized product configuration systems. In matters of information systems, the

same meta model is used for product description, matching and configuration [Wedekind, H. (2000), Hümmer, W. and Lehner, W. and Wedekind, H. (2002)].

2. PRODUCT CONFIGURATION WITHIN MASS CUSTOMIZATION

As mentioned above, the term "product configuration" is often used in the context of mass customization. Firstly, a product configuration system is the interface between a supplier and the demanders (product-related issue). In order to match supply against demand, the configuration system has to acquire the customer's needs and desires (such as size, color, alternative components etc.) based on the underlying product model. Secondly, these systems ensure that only valid product specifications can be chosen (technical issue).

As MC is a concept that can theoretically be applied to almost every branch or type of goods, there is no "one size fits all"-solution in matters of product configuration. For specifying the widespread meaning of MC, several classification models have been developed. An overview of theses classification systems is given in [Piller, F. T. (2001)]. It is suitable to have an overall term for what product configuration means in a particular MC scenario. But in fact, the requirements and functions of a product configuration system differ (e.g. individual cars in contrast to individual shampoos) [Wüpping, J. (2003), Wüpping, J. (2004)]. The reason for these differences can be found in the amount of customizable product parameters or in the production processes themselves. Another explanation for that is the intensity of the customer's involvement which can be called "customer integration". Several types of customer integration and its degree of intensity are shown in Figure 6.

System of Customer Integration	Interaction Point	Degree of Customer Integration
match-to-order / locate-to-order: Selection of existing (standard) products or services according to customer requirements	**sales, retail**	
bundle-to-order: Bundling of existing products/services to customer specific product	**sales, retail**	
assemble-to-order: Assembling of products/services from standardized components / process blocks	**final assembly**	
made-to-order: Manufacturing of customized products including component manufacturing	**manufacturing**	
development-to-order: Customer co-design of product/service, followed by customized made-to-order	**design, development**	

Figure 6. Customer integration [adapted from Piller, F. T. and Moeslein, K. and Stotko, C. M. (2004)]

In case of *match-, locate-, and bundle-to-order*, the customer becomes a partner of the value chain in the distribution process, i.e. at a downstream part of the value chain. Individualization for the customer is mainly achieved by selecting or packaging existing products (product components) at the point of sale. *Assemble- and made-to-order* means producing individual products with regard to the requirements of the customer. The customer partially cooperates within the fabrication process. Finally, *development-to-order* (resp. *engineering-to-order*) implies the highest degree of integration because the customer initiates development and production of a nearly new product. In contrast to a conventional batch production, efficiency potentials of the mass production will be included and combined [Piller, F. T. and Moeslein, K. and Stotko, C. M. (2004)].

Discussing production configuration systems for MC should take these differences into account. In this paper, we focus on match-, locate-, bundle-, and assemble-to-order. In the following section we will show that the integration of the data model and the personalized configuration system is essential for enabling these types of customer integration.

3. THE OVERALL PROCESS CHAIN

First of all we want to give a comprehensive overview of our overall process chain as depicted in Figure 7. The most important issue addressed by our approach is bringing together demand and supply of highly configurable products and

services as prevalent in MC. For this purpose we adopt a two step approach which is prototypically implemented by the Marrakesch Project [Hümmer, W. (2004)].

1. Match Making Step
   In the first step (Match Making) demand and supply are matched against each other on an electronic marketplace acting as an intermediary. The result of this phase is a set of possible product or service families close to the demander's needs. In the optimal case an exactly matching product offered by exactly one supplier is found. But in reality this is an illusion. Often demanders only have a vague idea of all characteristics and specifics the product or service has to provide. Therefore a second step, the final configuration is necessary.
2. Configuration Step
   What we have so far is a list of possible product and service families close to the demander's needs. These yield a set of possible configuration spaces the demander has to choose from. The demander has to decide which of these spaces are of interest to him (scoring). This activity has to be done by manual interaction. For the remaining supplies the demander has to enter an interactive negotiation process. We suggest a generic configuration system as the second step of our approach. The novelty of our configuration system is twofold: first we claim that the configurator can automatically be generated from the result of the match making step, i.e. our approach offers a *personalized configurator* leaving out those issues where demander and supplier have already agreed on. Second we allow demander and supplier to collaboratively and dynamically decide on the sequence of the remaining configuration activities.

In case of a successful configuration step the product or service is completely specified considering both the demander's needs within the supplier's portfolio. The following steps in the overall process chain, contracting and the execution of the contract, are well known and thus not further detailed here.

As we explained in the two steps above our approach is meant to minimize manual interaction. The fully automated match making step as well as the generation of the personalized configurator can only be achieved on the basis of a data model which can be used along the complete overall process chain. Further it has to be able to express highly configurable product and service families in a comprehensive and understandable manner.
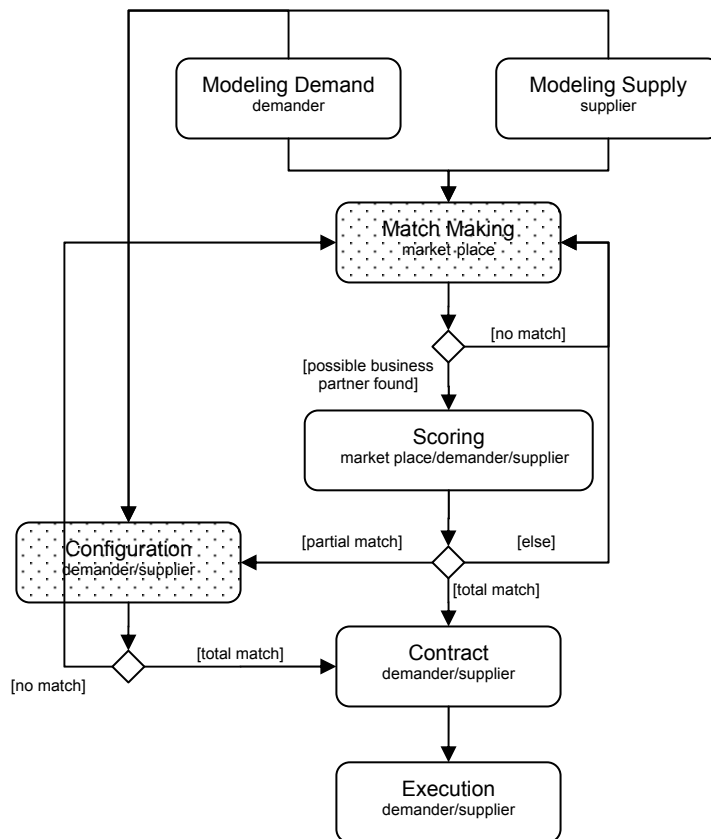
Figure 7. The Overall Process Chain

The rest of the paper is organized as follows: In the next section we introduce the data model by using a domain specific extension of the entity-relationship methodology. In Section 5 we consider both the matching step (5.1) as well as the personalized configuration step (5.2). After that we give a short overview over related work (Section 6) and close with a conclusion (Section 7).

4. THE MARRAKESCH DATA MODEL

The requirements the Marrakesch data model is based on are configurability, comprehensiveness and compactness. The data model also has to integrate into both the matching step as well as the configuration step. Further the requirements simplicity in the sense of usability and clarity are the motivation for a visual approach consisting on the one side of a graphical notation of the part-whole-relationship based on [Wedekind, H. and Müller, T. (1981)] and on the other side of classification trees based on generalization hierarchies [Jablonski, S. and Petrov, I. and Meiler, C. and Mayer, U. (2003)]. The basic modelling elements are: *Piece of Text* for final, not further decomposable parts, *Conjunctions* to express that several subparts are composed into a higher level part (this composition process can be done repeatedly) and *Alternatives* to express that an exclusive decision has to be made which subpart has to be chosen. To further extend the semantics of alternatives the *Empty Piece of Text* is introduced which allows the distinction between optional and mandatory alternatives. This enables us to express complex product or service models. But there are still two open issues: How to specify complex dependencies between alternatives and how to enable customization within specific ranges?

To efficiently restrict the domain of possible product variants specified by alternatives the concept of *Implications* is introduced. E.g. you can express that an eight cylinder engine implies an automatic gear box by using an Implication. This modelling construct leads to a reduction of complexity in our models and supports simplicity and clarity. Figure 8 gives an example of the benefit of Implications using an abstract use case that demands to choose the *Piece of Text* c, if a is chosen: both figures are equivalent with regard to the product family described, but the structure on the left is of lower complexity.

To enable different types of parameters we extend the Pieces of Text by typed attributes restricted by constraints. For example we can express that the size of a sneaker can be specified in inches but that the real number describing the length must be between 1 and 20 inches. The meta model illustrated in Figure 9 is based on a modelling approach specializing and extending the method specific ER-Model by domain specific extensions for mass customizable products [Jablonski, S. and Petrov, I. and Meiler, C. and Mayer, U. (2003). The content of the domain specific meta model (DSM, [Hümmer, W. (2004)]) expresses the concept of modelling products or services explained so far by deriving the necessary entities, relationships and attributes resp. constraints.

The method specific meta model (MSM) defines the different types of meta modelling elements and their relationships; in the case of ER the elements at disposal are Entity, Attribute, Domain and Relationship. Please note that Relationship on that level is an entity. The DSM comes into existence by specialising the entities defined by the types. In Figure 9 the special relationship *Part-Whole* can be seen, which is connected with the special entities *Sub Part* and *Upper Part*. The constraints imposed by this method of modelling can be exploited by modelling tools and support the overall modelling process.
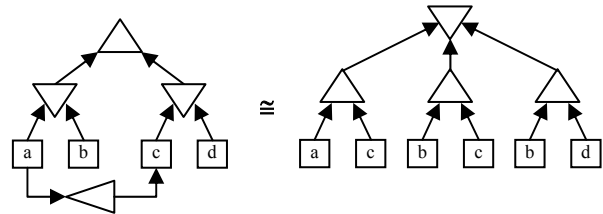
Figure 8. The Benefit of Implications

5. THE TWO-STEP APPROACH

5.1 Match-Making

To illustrate the matching step we use the example given in Figure 10. In the upper part the more complex supply of the PC family "Lightning" is presented consisting of three alternatives, two of which are connected by an implication. The implication expresses that choosing the 1800 MHz processor from A_1 makes it necessary to choose the active ventilation system from A_2. In the lower part of the figure the demand is depicted requesting a PC system containing a special kind of graphics adapter as well as a fast processor with at least 1500 MHz. Due to the simplicity of this example it is obvious that the demand can be satisfied by the supply presented above. In the following we give a short outline of necessary steps for matching.
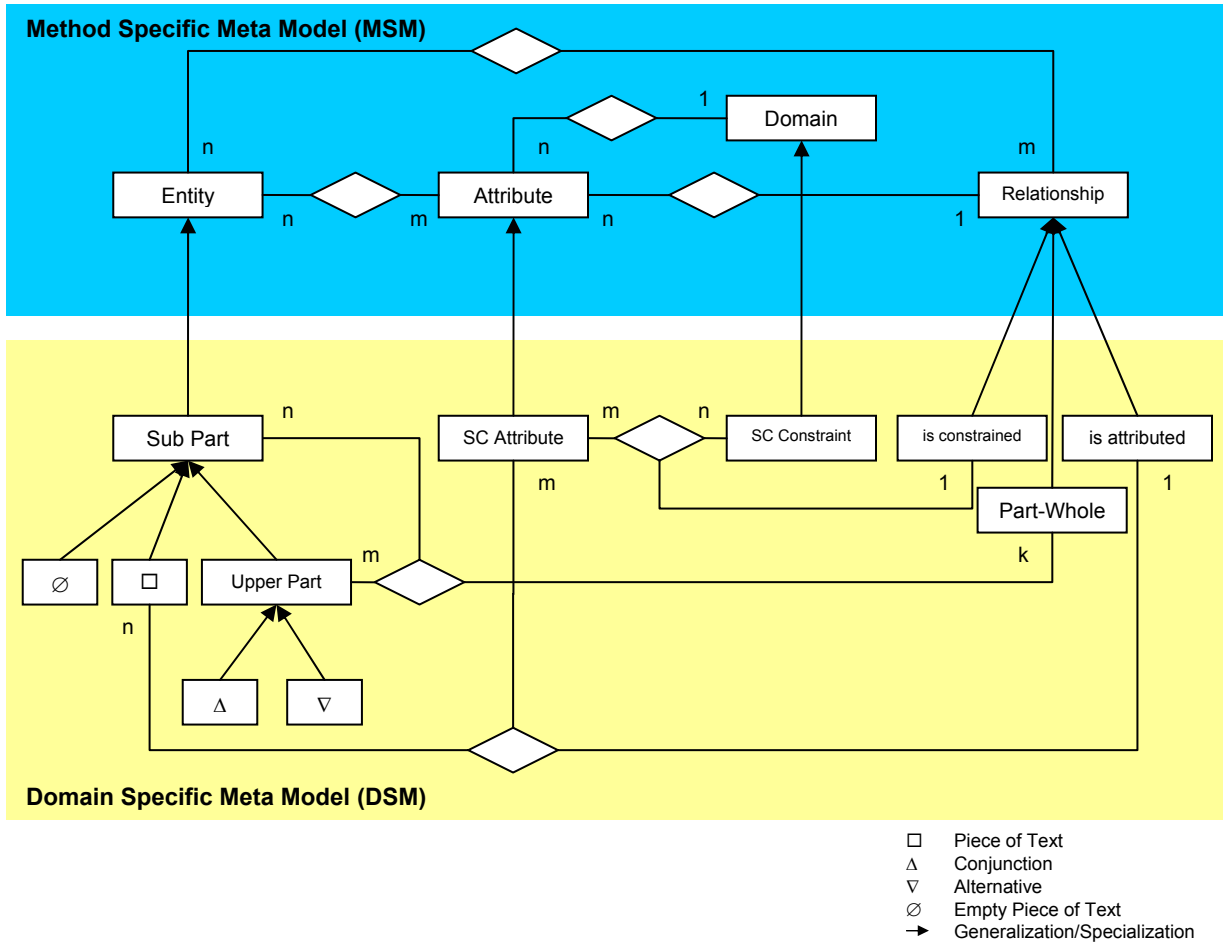
Figure 9. Meta model for mass-customizable products and services

Within a loop over all pieces of text of the demand structure each of the pieces of text has to be matched against a node of the supply structure. It is interesting to note that the matching node of the supply side is not necessarily a piece of text but can also be a complex bill of materials rooted by an alternative or conjunction. This demonstrates once more the advantages of the chosen data model because it allows matching very exact specifications on the one side against rather vague ideas on the other side. The matching is computed by comparing the position of the nodes inside the data model graphs as well as the contents of the nodes. In our example first of all a matching part from the supply side is needed for ts'_1, the special graphics adapter; ts_2 is returned as the matching node.

Then a matching node for ts'_2, the processor, is needed. The node matching based on the predicate 'frequency $\geq$ 1500 MHz' yields ts_3, the 1800 MHz processor. As ts_3 is on the one hand the subpart of an alternative and on the other hand the starting point of an implication, special actions have to be applied: first the alternative A_1 can be resolved, i.e. it can be removed with all the siblings of ts_3 from the supply structure by attaching ts_3 directly to the top conjunction K_1. Further the implication is activated thus automatically deciding A_2: To further guarantee the validity of the supply configuration ts_5 has to be chosen and A_2 can be resolved in a similar manner as A_1. As the demand structure contains no further nodes the matching step comes to an end. The result is the reduced supply structure depicted in Figure 11. Obviously the alternative A_3 is not decided yet.

We are aware that the matching problem is NP-complete as it is closely related to the tree-to-tree-matching problem ([Selkow, S.M. (1977)]). The problem is complicated further when regarding the large number of participants of an electronic market place. Therefore it is not reasonable to tackle real world configuration problems with classical deterministic algorithms. As part of the Marrakesch prototype several kinds of heuristic algorithms for efficiently solving the matching problem were developed. Possible strategies are based on simulated annealing or genetic computation, but are out of scope of this paper. For details see [Hümmer, W. (2004)].

K₁: conjunction PC "Lightning" – supply

ts_1:   memory
ts_2:   graphics adapter
A_1:   alternative frequency
        ts_3: 1800 MHz
        ts_4: 1000 MHz
A_2:   alternative active ventilation
        ts_5: active ventilation
        NULL: no additional ventilation

41

K₁: Konjunktion PC "Lightning" (Angebot)
ts₁: Speicher
ts₂: Grafikkarte
A₁: Alternative Prozessortakt
  ts₃: Takt = 800 MHz
  ts₄: Takt = 300 MHz
A₂: Alternative Aktivlüfter
  ts₅: Aktivlüfter
  Null: kein zus. Lüfter
A₃: Alternative Brenner
  ts₆: CD-Brenner
  Null: kein Brenner
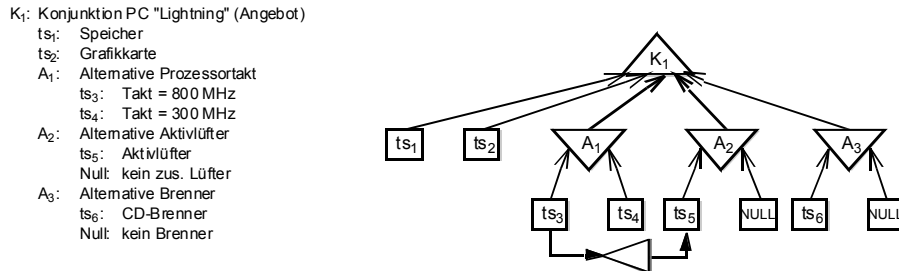
a) supply: configuration scheme of a computer system

K'₁: conjunction PC (customer's demand)
ts'₁: graphics adapter xy
ts'₂: frequency >= 1500
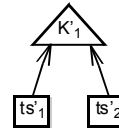
b) demand specification of a computer system

Figure 10. Demand and supply

5.2 Personalized Configuration

So far we have successfully executed the matching step which is the first of our two step approach. The result of this step as illustrated by the example above is a reduced supply structure fulfilling the demander's needs but still containing certain open issues, as A_3, the CD writer option in Figure 11. In essence the configuration task consists in resolving the remaining undecided alternatives of the supplier's reduced data structure. With this observation in mind our market place system is able to automatically generate a personalized configurator. Personalized within this context means that the customer is given the possibility to choose from a range of products that all fulfill his primary requirements, i.e. none of the products or services the personalized configurator presents violates his needs.

Figure 11. Possible result of the matching process

The personalization approach is not the only benefit of our solution; additionally we offer a second degree of freedom of choice concerning the sequence of the configuration steps. So two questions arise:

- How is a single configuration step, i.e. the decision of an alternative, achieved?
- How is the sequence of the configuration steps computed?

These questions depend on each other because the answer to the second question for the sequence depends is the basis for the first question for the alternatives. Thus we reach a dialog on two levels:

- To decide the alternatives we introduce the *Content Dialog*. This kind of dialog is to embody a single configuration step, i.e. the customer has to choose from the possible options of the current alternative.
- To determine the sequence of the content dialogs we introduce the concept of the *Meta Dialog*. This allows the customer together with the configurator to configure more important issues earlier than other issues.

By distinguishing content and meta dialogs we achieve an overall configuration process as depicted in Figure 12. The starting point of the second step of our two step approach is the meta dialog because first of all a starting sequence has to be determined, i.e. the first alternative to be configured has to be chosen. If the meta dialog cannot result in a sequence according to the demander's requirements he can choose to abort the meta dialog and by that the whole configuration process. If the demander and the configurator can agree on a sequence they proceed to the first content dialog about the first alternative of the determined sequence. Again this dialog can lead to the overall abortion. If the dialog is successful the current configuration based on the decisions of the alternatives taken so far has to be checked. This can be done fully automatically by the configuration system. The Check Configuration process on the one hand has to reduce the variant space provided by the data model with regard to the outcome of the previous content dialog. This can possibly result in the reduction of the remaining set of alternatives to be decided. Further by deciding an alternative within the content dialog it has to be checked if an implication is fired. In that case further alternatives might be decided or become obsolete. On the other hand the Check Configuration process also has to find out if the configuration is complete, i.e. all alternatives of the
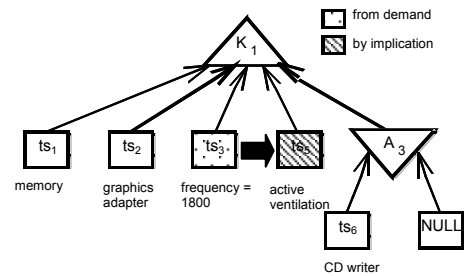
data model are decided. Then the configuration process is finished and the demander can go on to the next step in the overall process chain.

Now we proceed from this high level point of view to a more detailed description of the Meta and the Content Dialog representing the main issues of the configuration process. The internal structure of the meta dialog is depicted in Figure 13. The dialog starts with a possibly empty proposal of a sequence *seqp*. If this initial parameter is not empty the demander can decide if he wants to skip the meta dialog, i.e. he is still satisfied by *seqp* generated in an earlier meta dialog. If the proposal is empty or the customer wants to reschedule the configuration sequence the core of the meta dialog has to be entered. First of all it has to be decided who will get the right of proposal, i.e. who is allowed to suggest a configuration sequence. Depending on the general environment of the configuration system the customer may be awarded the right of proposal otherwise the configurator has to generate a sequence. The result in both cases is a new

Figure 12. The overall Configuration Process

proposal of a sequence *seqp'*. The opposite party has to decide whether it agrees to this proposal or not. In the negative case the demander can chose to repeat the meta dialog. This option on the one hand is very beneficial and realistic, because in the next iteration a more suitable sequence might be found. On the other hand the repetition bears the risk of entering an infinite loop. If a settlement is achieved the demander can proceed in the configuration process with the sequence proposal *seqp*.

Figure 13. Schema of a Meta Dialog

Now the content dialog (Figure 14) on the first alternative *alt* of the current sequence proposal *seqp* can be started. After extracting alt from *seqv* the state of *alt* has to be checked. Because of validation actions and the firing of implications it can happen that *alt* is already decided or not relevant to the current configuration. In that case the next element from the sequence proposal has to be fetched. A repeated fetching loop can result in an empty current alternative *alt* which makes it necessary to leave the content dialog without changing the configuration and to start a new meta dialog on the sequence to be pursued. If finally an alternative is found that has to be configured the real configuration step can be approached. With regard to our data model presented in Section 4 the possible options to be configured are the still available subparts of the alternative. Some of these options from the original product family might not be available any longer because of validation reasons. If the customer can find an option suiting his needs a settlement is reached and the current data structure *MG* has to be transformed accordingly to *MG'*. This transformation only consists of removing the options not chosen. All further modifications of the data structure are left to the Check Configuration process depicted in Figure 12. In case the demander cannot find an appropriate option he is enabled to abort the current content dialog and by that aborting the whole configuration process. Content dialogs as explained have to be repeated with possibly intermitting meta dialogs until a single configuration within the configuration space expressed by the data model is achieved.
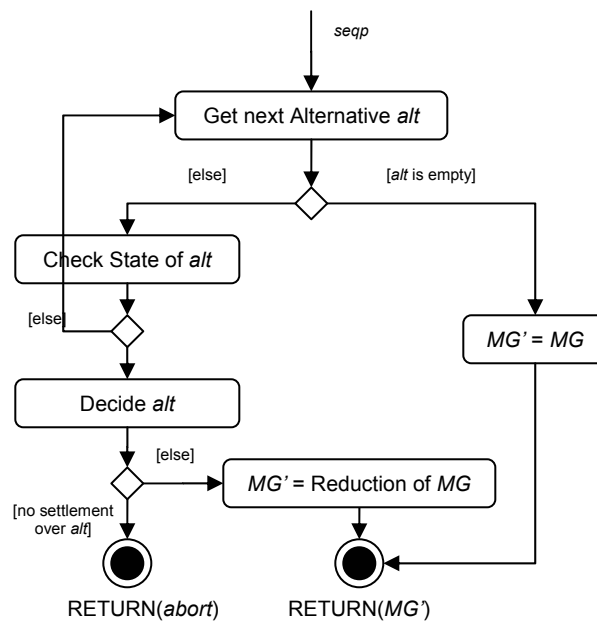
Figure 14. Schema of a Content Dialog

6. RELATED WORK

The term "product configuration" in this paper is seen as a major part of the global MC process. Most of work related to our approach has its focus on configuration technologies and product models. Wüpping illustrates different types of configuration methods (such as a rules-based and knowledge based approach). Organizational and technical aspects of implementation are shown [Wüpping, J. (2003), Wüpping, J. (2004)]. Hahn proposes a common model for product information (e.g. product catalogue) and configuration rules by using decision trees [Hahn, A. (2003)]. Hvam et al. present a procedure for building product models. For product analysis and object oriented analysis as two phases of the procedure trees and UML class diagrams are used [Hvam, L. and Riis, J. and Malis, M. and Hansen, B. (2002)]. A General model of product families is illustrated in [Ying-Lie O (2002)]. Model-based configuration is described as a combination of product modeling and solving a configuration problem. [Wimmer, A. and Mehlau, J. I. and Klein, T. (2003)] introduce a meta model for object oriented product modeling. This model has been developing for banking industry in order to describe financial products. It consists of the product meta model and the product model itself. A conceptual configuration model is described in [Felfernig, A. and Friedrich, G. and Jannach, D. (2001)]. Because of the increase of product and process complexity the need for effective knowledge acquisition enforces more and more. For modeling configuration knowledge bases UML is used as it is a approved standard design language. Janitza et al. have developed a meta model which intents to combine a customer, product development, and production perspective. The both main parts of the UML notated model

are a product spectrum model (characterizing the whole solution space of valid products) and a product configuration model (as instance of the product spectrum model with a specific product configuration) [Janitza, D and Lacher, M. and Maurer, M and Pulm, U. and Rudolf, H. (2003)].

The novelty of our approach presented in this paper is twofold. First of all, we combine the static Marrakesch data model which is of fundamental importance for specifying mass-customizable products with the dynamic over-all process chain. Secondly, by introducing the personalized configuration we reduce the complexity of the configuration task for the customer.

7. CONCLUSION

In this paper, a two-step approach for integrating technical and organizational issues of MC product configuration was presented. We specified the underlying definition of Mass Customization as a management concept and focussed on different types of customer integration. Then, the overall process chain with its two main steps was introduced. Match Making as the first step builds a set of possible product or service families that fit the demander's needs. The second step called Configuration enables to build a personalized configurator, which is used to decide on the remaining issues that could not be resolved by the Match Making step.

The objective of this paper is to merge technical and organizational aspects. This seems challenging because successful MC projects have to ensure that offers can be realized in an effective and efficient way for the supplier on the one hand and product configuration on the other hand has to fulfil each customer's needs. Therefore, we point out that different types of MC resp. different types of customer integration imply varying product configuration concepts. In this paper, technical aspects like matching supply and demand as well as modelling product structures using a meta model are shown. From the organisational point of view we suggest to involve the customer into the process of establishing new product structures, the integration of the customers' requirements in adjusting existing product structures (knowledge management), creating personalized configurators based on the chosen solution space for the desired goods and specifying the interaction processes among a configuration system and a user (mostly a customer).

Further work has to be done on cooperatively designing product structures, supporting product requirements engineering for adjusting configuration options and evaluating case studies of several domains to test the transferability of this approach.

8. ACKNOWLEDGEMENTS

8. REFERENCES

1. Piller, F. and Stotko, Ch. M. (2002). Four approaches to deliver customized products and services with mass production efficiency. In: Durrani, T. S. (ed.): Proceedings of the IEEE International Engineering Management Conference (IEMC 2002). Managing Technology for the New Economy. Cambridge University, pp. 773-778.

2. Dietrich, A. and Hümmer, W. and Meiler, C. (2004). A Meta Model based Configuration for mass-customizable Products and Services. In: Proceedings of the 4th Workshop on Information Systems for Mass Customization (ISMC 2004), Madeira, Portugal.

3. Wedekind, H. (2000). On Specifying Contract Negotiations. in: Hansen, H.-R. et al. (eds.): Proceedings of the 8th European Conference on Information Systems (ECIS2000), Volume 1, Vienna, pp. 23-30.

4. Hümmer, W. and Lehner, W. and Wedekind, H. (2002). Contracting in the Days of eBusiness. Sigmod Record 31/1, pp. 31-26.

5. Piller, F. T. (2001). Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter. Wiesbaden.

6. Wüpping, J. (2003). Konfigurationstechnik für die individuelle Serienfertigung. IT Management 7 (2000) 4, pp. 2-9.

7. Wüpping, J. (2004). Produktkonfiguration – Der Weg zur kundenindividuellen Serienfertigung. see: http://www.technikwissen.de/library/content/17/2001/03/152.pdf. access: 2004-06-14.

8. Piller, F. T. and Moeslein, K. and Stotko, C. M. (2004). Does mass customisation pay? An economic approach to evaluate customer integration. In: Journal of Production Planning and Control (to appear).

9. Hümmer, W. (2004). Marrakesch. http://www6.informatik.uni-erlangen.de/research/projects/marrakesch/. Access: 2004-06-14.

10. Wedekind, H. and Müller, T. (1981). Stücklistenorganisation bei einer großen Variantenzahl. In: Angewandte Informatik 9, pp. 377-383.

11. Jablonski, S. and Petrov, I. and Meiler, C. and Mayer, U. (2003). Metadata Repositories as Infrastructure for Database Oriented Applications Published. In: Proceedings of The 15th Conference on Advanced Information Systems Engineering, Caise Forum, CAiSE 2003, Velden, Austria.

12. Selkow, S.M. (1977). The Tree-to-Tree Editing Problem. In: Information Processing Letters, 6(6).

13. Hümmer, W. (2004). Vertragsverhandlungen um konfigurierbare Produkte im elektronischen Handel (Contract negotiation on configurable products in electronic business). Dissertation, to appear.

14. Hahn, A. (2003). Integriertes Produktkatalog- und Konfigurationsmanagement. in: Industrie Management 19/1, pp. 29-32.

15. Hvam, L. and Riis, J. and Malis, M. and Hansen, B. (2002). A Procedure for Building Products Models. In: Rautenstrauch, C. and Seelmann-Eggerbert, R. and Turowski, K. (eds.): Moving into Mass Customization. Information Systems and Management Principles. Berlin, pp. 19-39.

16. Ying-Lie O (2002). Configuration for mass-customization and e-business. in: Proceedings of the Configuration Workshop at The 15th European Conference on Artificial Intelligence ECAI, Lyon, France, pp. 11-16.

17. Wimmer, A. and Mehlau, J. I. and Klein, T. (2003). Object Oriented Product Meta-Model for the Financial Services Industry. Proceedings of the Mass Customization and Personalization Congress 2003. Munich.

18. Felfernig, A. and Friedrich, G. and Jannach, D. (2001). Conceptual modeling for configuration of mass-customizable products. In: Artificial Intelligence in Engineering 15 (2001) 2, pp. 165-176.

19. Janitza, D and Lacher, M. and Maurer, M and Pulm, U. and Rudolf, H. (2003). A Product Model for Mass Customisation Products. In: Palade, V. and Howlett, R. J. and Jain, L. (Eds.): Proceedings of the 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES 2003, Part II. Oxford, UK, Sept, pp. 1023-1029.

Development and implementation of product configuration systems – a change management perspective

By Morten Møldrup (corresponding author) and Niels Møller

Department of Manufacturing Engineering and Management, Technical University of Denmark, Matematiktorvet, Building 303, 2800 Kgs. Lyngby, Denmark (email: mmd@ipl.dtu.dk).

Abstract

Product configuration systems (PCS) are IT-based expert systems, which contain product information and knowledge and in turn provide the basis for automating the decision-making processes in the order acquisition phase. Thereby PCS support and streamline the efforts of an organisations sales force in translating customer demands into customized products within a bounded range of product variation. Supported by a PCS the sales representatives are empowered to configure products without having to consult experts about product specific dependencies.

Although several research projects have been carried out within the field of PCS, these efforts have primarily been focused on technical aspects. Based on experiences gained from twelve case studies in the context of the PETO (*) project, this article therefore sets out to broaden the scope of the PCS field to embrace managerial and organizational aspects and understandings as well.

Within managerial and organizational research studying information systems in general and expert systems in particular the degree of success in developing and implementing such systems is known to depend closely upon the project team in charge and their capability of understanding, navigating in and acting upon the social context in which the development of PCS takes place. Especially dealing with resistance towards change among experts and users is a core activity to handle carefully when planning and carrying out such projects.

Therefore the focus of this paper is the employment of a change management perspective with emphasis on preventive initiatives taken to deal with possible expert and user resistance. In this respect the analysis reflects that the actual practice of change management in the twelve companies to a large extent is based purely on past experience and common sense. But although the actual practice therefore is implicit and non-formalised, these change management related considerations and actions are still found to contribute to the overall successes of the PCS projects.

Thereby this paper brings the research community in the field of PCS as well as existing and future adopters of the PCS technology a step further in understanding, which managerial and organizational reflections the PCS projects call for in order to enhance the likelihood of success in developing and implementing PCS. Conclusively we state that project teams involved in PCS projects can enhance their likelihood of success by paying a stronger attention to the notion of change management.

Keywords: product configuration, expert system, information system, change management, user participation, expert resistance.

Introduction and theoretical methodology

Organizations have in the past thirty years at an accelerating pace invested vast amounts of resources in developing and implementing IT-based information systems (IS). While some IS such as enterprise resource planning systems (ERP) and Intranets are broad in their scope, others such as customer relationship management systems (CRM), human relationship management systems (HRM) and management information systems (MIS) are dedicated specific departments or functions within organizations.

Alongside the domain of IS the field of expert systems (ES) has emerged from Artificial Intelligence research and has been established as a self-contained research area as a subset of the broader field of IS. The development and implementation of ES aims at retaining valuable expert knowledge within an organization thereby making it readily accessible for every employee that need this knowledge in order to perform well or better in their different job functions. Thus ES are frequently

developed and used as effective tools assisting and advising less knowledgeable employees in routine and repetitive problem solving and decision-making activities without having to consult experts (Coakes and Merchant, 1996, Lightfoot, 1999).

Although ES have been widely criticized for not living up to the expectations as *"revolutionary solutions to problems in almost every area of human activity"* (Simons, 1988 in Coakes and Merchant, 1996) thereby not providing the proclaimed benefits, the efforts of retaining expert knowledge remain and have in the 90´s found its way into the field of product configuration. As such PCS are tools supporting the efforts of companies trying to adopt mass customization bearing in mind that the *elements of mass customization require automation tasks where it makes sense* (Pine, 1993). Thus industry has in this case in fact found applicability for ES despite of previous evidence suggesting *"...that expert systems remain as academic concepts rather than being used as tools for industry"* (Lehaney et al., 1993).

A product configuration system is basically a computerized model of a product, which allows a sales representative or customer to choose (configure) a desired composition of a particular product through a structured sequence of choices. The PCS contains information about the individual components and expert knowledge about which components are allowed to be joined thereby supporting and facilitating the decision-making process of the user. If a sales representative was to configure an automatic coffee maker, the PCS would for instance automatically ensure that the size of the coffeepot and the capacity of the heating element match the specific coffee volume. Thus the user is by means of the PCS capable of narrowing the immense number of options down to one or a few product variants, which fulfil the desired product requirements.

In this respect the purpose of a PCS is to automate a substantial amount of the time-consuming iterations back and forth between the sales representative and various other departments such as product management, R&D and production, which till now have had to be involved in specifying the customized products. Hence the sales representative can engage in ongoing dialogue with a customer without having to consult experts for advice and guidance as long as the desired product falls within the boundaries of the PCS.

The persisting growth of IS and ES mean that these systems now pervade and influence almost every aspect of organizational life, but this trend has been far from unproblematic. For instance Jiang et al. (2001) states that the *"failure rates of large-scale systems development projects are extremely high, with many organizations either abandoning their efforts after allocating substantial resources or failing to achieve the anticipated outcomes from their investments"*.

The success rate of projects aiming at introducing new IS into organizations has previously been found to be dependant on a large number of variables. In this context a vast number of IS projects have failed due to the lack of focus or inability of project teams to include and address the social issues, which are interweaved technical issues throughout the project. To name a few of the main and tangible characteristics of PCS projects we can for instance mention the creation of new organizational entities such as product configuration teams and departments and considerable re-design of the work processes of sales representatives as well as routines and procedures concerning documentation of product knowledge.

Existing literature and the above examples thus indicate that the successful development and implementation of PCS requires careful attention towards the considerable amount of organizational changes, which accompany PCS projects. These changes affect a large number of employees at all levels throughout the company, who in turn have to make *"the needed sacrifices"* as change is never painless (Kotter, 1996). And therefore we expected the concept and practice of change management to be widely covered and elaborated within the field of IS research. But the review of a substantial amount of literature revealed that although strong emphasis on change management related issues such as risk analysis and user resistance towards IS were in fact present, the IS literature to our knowledge only sporadically utilizes the concept of change management more systematically.

Based upon this observation we claim that the use of a more coherent change management framework will be advantageous in order to analyze and explain why some IS projects such as PCS projects are more successful than others, as it is widely known that employees seldom embrace change without reservations, concerns or actual resistance towards change.

As PCS are planned by the members of the companies at hand these being early adopters of the product configuration technology in order to solve problems, adapt to external environmental changes and improve performance, we have sought inspiration in the notion of planned change. Theories and approaches within the field of planned change have been conceptualized in a variety of phase models or frameworks that describe the activities, which must *"take place to initiate and carry out successful organizational change"* (Cummings and Worley, 2001).

When analyzing the empirical data obtained from twelve Danish companies we in particular found Kotters (1996) eight-stage change framework useful, as it provides an oversight as well as particular points of reference to consider, when seeking to change organizations. On that basis we apply substantial parts of this framework as a recurrent and guiding element throughout this paper in order to analyze the actions and initiatives carried out by project teams and project

managers. However, the emphasis has been put the first six steps these being: *1) Establishing a sense of urgency, 2) Creating a guiding coalition, 3) Developing a vision and strategy, 4) Communicating the change vision, 5) Empowering employees for broad-based action* and *6) Generating short-term wins*. The reason why is that none of the twelve case companies were able to provide any substantial amount of data in relations to the last two steps: *7) Consolidating gains and producing more change* and *8) Anchoring new approaches in the culture*. The central reason why is that the majority of the companies have not gained any experiences with respect to the last two steps within the timeframe of the PETO project, as the PCS projects are still running.

But as the framework developed by Kotter (1996) is meant as a generic and therefore generalized model when planning change projects, it lacks the subtlety needed in order to fully comprehend the more specific and interrelated problems at hand in the development and implementation of PCS. Therefore we have brought other theoretical perspectives into the analysis to shed further light onto the change management issues at hand.

In this respect it has from different viewpoints been argued that user involvement and participation is an important issue in developing IS ensuring everything from workplace democracy to capitalizing on the huge investments that go into IS projects. Among other user related aspects influencing IS success such as failing to meet user's needs (Gallagher, 1998 in Jiang et al., 2001), unwilling users and multiple users (Anderson and Narasimhan, 1979 in Jiang et al., 2001) and the inability to specify purpose or usage, the inability to cushion the impact on others and the loss or lack of support (Alter, 1979 in Jiang et al., 2001) especially user involvement has been dragged forward as a main issue to be aware of. The reason why often being that user involvement and participation is the key to design sustainable workplaces and handle user resistance towards the IS.

Following this lead user involvement is according to Agarwal and Prasad (1997) essential, because the organizations do not realize the promised benefits unless the IS is in fact used. Identifying users perception and beliefs according to IS as a key variable in understanding users decisions to either abandon or in fact accept and adopt the IS, Agarwal and Prasad (1997) go on by suggesting that user characteristics such as personal innovativeness might be a moderating influence linking user perception and acceptance of IS. Agarwal and Prasad (1997) also shed further light into the relative efficacy of different communication channels in relations to influence user perception.

Recognizing that ES are knowledge based thus different from IS in general, the same risks concerning user resistance also apply for ES. But due to difference mentioned above the change management efforts in developing and implementing ES also have to take into account, how experts are motivated to participate willingly in the PCS project. Needless to say this is of major importance as valid expert knowledge is an essential prerequisite, when developing ES, because this knowledge constitutes the knowledge base of ES (Lightfoot, 1999).

In that respect quite a few research studies emphasize that ES projects will probably fail, unless experts within the organization are willing to cooperate (Lightfoot, 1999). Acknowledging that many ES projects set out with the purpose of replacing the very same experts, who are to contribute to the project by placing their knowledge at the disposal of the project teams (Coakes and Merchant), this area is to gain considerable managerial attention when developing ES. Especially since Turban (1992) in Lightfoot (1999) argues that unwilling experts contribute to the striking failure rate of 85% to 90% with regards to ES.

Altogether managing and organizing the development and implementation of ES such as PCS seem to comprise more resistance related risks than IS in general, because the project team has to deal with expert as well as user concerns, reservations and scepticism. All in all this talks for project teams that besides managing a vast number of other and interrelated technical and social aspects are capable of understanding, acting upon and thus overcoming resistance among these key employees in order to be successful.

Founded in the above background this article sets out to analyze and reflect upon experiences gained from the PETO project with a primary focus on the issues of expert and user resistance within the domain of PCS by applying generalized change management research as well as prior research findings with regards to the development and implementation of IS and ES.

Next we shortly explain the empirical methodology used in this paper after which we turn to the analysis and discussion of the empirical findings in the PETO project. This section of the paper is divided into six subsections, which reflect the first six stages in Kotters (1996) change management framework. As a framework can never be more than a "model of the reality" these subsections to overlap to some extent, and you might find passages in one subsection, which you correctly would anticipate to be found elsewhere. However, we believe that this structure supports our claim in this paper. Last you will find the conclusion.

Empirical methodology

This research was conducted in the "The Product Configuration Systems, Economy, Technology, and Organization" project (PETO), which was formed to study the process of and effects by developing and implementing product configuration systems in a variety of Danish companies.

A number of twenty companies were approached of which twelve agreed to participate in the study. In this respect the vast majority of the twelve participating companies have been fairly successful in developing and implementing PCS i.e. they have developed PCS, which are currently being used by internal and/or external users.

At the beginning of the PETO project we tried to include a number of less successful companies in the PETO project in order to get a wider scale on which to analyze, why some projects failed while others were successful. But as the less successful companies in spite of our persistent efforts either refused or kept dragging their acceptance on and on, this article is based almost solely on fairly successful companies. Still we find that the twelve investigated companies constitute a sound basis of successful and less successful experiences on which to extract and elaborate upon, as none of participating companies came through without experiencing some degree of change management related problems. In that respect a number of conspicuous examples will be extracted from the case studies in order to accentuate, illustrate and concretize the applied change management framework in a PCS context.

Although judging the degree of IS success is a complicated matter pointing to a vast variety of relevant quantitative indicators such as project delivery within the scheduled timeframe or keeping budgets as well as qualitative points of reference such as easing the daily work of users, we base our judgment on a somewhat simplified foundation. That is, PCS project are evaluated as successful, if the project team through managerial and organizational initiatives overcomes the issue of expert and user resistance and in fact delivers a PCS, which is accepted or in fact well liked by these two essential groups of employees. Accordingly, we take the perspective of these stakeholders thereby not dealing systematically with benefits for the companies nor the industry or the nation, which could have been other perspectives to adopt (DeLone and McLean, 2002).

In order to gain a broad understanding of the impact of PCS the case companies were asked to provide respondents in the following categories: 1) Sponsor, 2) Technician/programmer, 3) User, and 4) The project manager. These four roles were chosen, as they were expected to represent the most significant roles in a configuration project (Riis, 2003) bearing in mind that at least one of these persons also represent the role of the expert. Based upon the first draft of a questionnaire including economical, technical and managerial/organizational questions a number of semi-structured interviews were carried out at one case company. Afterwards the questionnaire was revised and used in the other cases.

Due to easy access to the company mentioned above, additional interviews were possible in order to follow up on questions that arose during the first analysis. As the empirical data from this case is most exhaustive, a more detailed analysis was made possible and in turn findings from this case are relatively stronger represented in this paper than are the rest of the cases. For a more comprehensive exposition of the empirical methodology see for instance Edwards and Møldrup (2004).

Analysis and discussion

Establishing a sense of urgency

Kotter (1996) argues that the far most important issue when trying to foster change within an organization is the establishment of *"a high enough sense of urgency in fellow managers and employees"* in order to erode potential complacency with the status quo. In other words, if especially key groups, whose support throughout the PCS project is important, do not acknowledge the need for change, a potential change initiative is doomed to failure, even before it has lifted off. Hence establishing a sense of urgency to change throughout the organization must never be underestimated (Kotter, 1996).

In quiet a few cases the PCS projects were initiated due to initiatives from outside the company and before proving the technology and demonstrating the potential benefits these projects at first lived a more or less secluded life in the organization. In other words the change process started with just two or three people, which is quite common (Kotter, 1996). Nevertheless companies being successful in developing and introducing PCS into the organization were characterized by an already existing understanding and acknowledgement of the need for change. In these cases the initial discussions among employees, typically holding management positions, concerning the project focused on recognizing and prioritizing the problematic issues in the quotation and order handling processes and further aligning and integrating these considerations into the strategic context of the organization. Thereby the broad business performance and not narrow functional goals (Kotter, 1996) were guiding the successful PCS project.

Moreover these companies were in this phase often assisted by researchers from the Technical University of Denmark with prior experience with the development of PCS thereby further qualifying the discussions and providing valuable insight into areas of interest, which had not been considered by the company representatives. In some cases ph.d.-students also did thorough mapping of the quotation process, which brought them into contact with a vast number of experts and future users

thus gaining additional insight into and influencing the thoughts and considerations of employees and at the same time not burdening the project budget as their salaries were already covered by the university. In this respect these companies obtained slack resources, which are agreed upon as being beneficial in innovative and adventurous projects (Keegan and Turner, 2002) such as PCS projects.

Although problems as long lead times, poor quality of specifications etc. were in fact main, through-going issues among the companies at hand (Edwards and Riis, 2004) several companies also used company specific reasons in establishing the sense of urgency. For instance one case relayed strongly on the immanent need to retain expert knowledge due to an elderly workforce, which is about to reach retirement age shortly after implementation of the PCS.

And another case company developed a PCS alongside the development of a highly complex product, which incorporated existing as well as newly developed components and modules into an overarching product solution. Due to the acknowledgement that the sales force did not possess the necessary knowledge nor time to individually specify every product for the specific customers, the R&D department understood the sense of urgency and willingly shared their knowledge with the knowledge engineers in charge of developing the PCS.

Developing a vision and strategy

As a result of the preliminary discussions almost every company did develop a vision and a strategy for how the vision was to be achieved (Kotter, 1996), and during the initial phases these first drafts evolved. Essentially a desired future was described including a scope and a prioritized list of requested and to some extent measurable benefits to be gained from the PCS project. For instance the PCS was in one company an essential mean to make a strategic shift in the market selling off the production and focusing on assembling thereby *"conveying a picture of what the future will look like"* (Kotter, 1996). It should however be noted that the development and implementation of PCS in many cases was part of a bigger project for instance focusing on the introduction or renewal of ERP systems. See also Edwards and Riis (2004) for an analysis of expected and realised benefits when implementing PCS.

The twelve case companies investigated demonstrated that the project team continually learned about additional barriers and opportunities, many initial visions were found not to be sufficiently clear to provide appropriate guidance for the project team throughout the project. Not having the advantage of a clear cut vision, a lot of meetings in the project team and time in general were spent with *"torturous discussions"*(Kotter, 1996) re-visiting different issues such as project boundaries, level of product detail and defining which constraints in the PCS should be hard or soft thereby re-opening already agreed upon decisions.

We find the notion of torturous discussions relevant in respect to visions that for instance do not state the reasons why employees affected by the development and implementation of the PCS should strive to create this future. On the other hand Keegan and Turner (2002) emphasize the need for committing the appropriate amount of time to investigate the complexity and understanding the issues at hand, when organizations venture into domains formerly unknown to the organization. This indicates that many a discussion referred to as *"torturous"* by (Kotter, 1996) might indeed be rational and sensible investments after all, as a bias towards the practise of rationalized project management tend to stifle innovative projects causing them to fail (Keegan and Turner, 2002).

Experiences from the twelve companies also support our observation that the balance between on the one hand driving the project forward within agreed time frames and budgets and on the other hand allocating slack resources to the project allowing the projects teams to do thorough investigations into the technical and social matters at hand as well as the relationship in between is very difficult to master.

Not having the internal resources necessary one case company decided to hire an external project manager, who to a large extent was to manage the project from a distance although frequent and lasting visits were in fact made to the company. A highly visible consequence of this arrangement was unclear responsibilities and that the project from time to another seemed not to make any progress, which in turn made the CEO apply additional pressure on the project team in order to meet deadlines.

Interviewing other key actors in this project revealed that this approach made the project team rush along without fully understanding the complexity at hand. For instance key actors stated that too little time was used to comprehend the knowledge management toolbox provided by the Product Modelling Group at Technical University of Denmark and understand the product knowledge in depth. In retrospect this meant that a too complicated product family for the project team to handle within the resources allocated was chosen. Also programming was forced through, and the project was closed before documentation was in place meaning that the PCS cannot be updated without external help from one specific, external consultant.

In this respect a key actor in another company emphasized that the fact that the steering committee was being very patient with the project team i.e. allowing several readjustments of the timeframe, this being a highly contributing factor in terms

of the overall success of the PCS project. And although important lessons were learned qualifying future PCS projects, a third case company had to abolish the first PCS project, because not sufficient time was taken to fully understand the socio-technical nature of the project, this resulting in a too strong bias towards technical issues at the expense of organizational issues such as the context in which the PCS was to be implemented.

Based on these findings as well as Keegan and Turner (2002) we recommend that when the PCS technology is new to an organization, sufficient time is spent in the early phases of the project to gain the needed insight into the technology and overview of the product structure. We believe that this approach will reduce the need to revisit discussions frequently in later phases of the project. However, we must admit that too little emphasis was put on collecting data during interview, which could have introduced further light and shade into the analysis of which discussions were necessary and which were torturous.

Creating the guiding coalition

As major change can hardly ever be accomplished by the effort of a single individual, Kotter (1996) advocates the creation of a *"guiding coalition"*. Using Kotters (1996) four key characteristics for designing an effective guiding coalition; *"position power, expertise, credibility and leadership"*, we now focus the attention to some of case companies at hand and how they went about the issue of organizational project design.

Although especially smaller companies had several key players taking part in as well strategic, tactical and operational tasks, the majority of the successful companies in the PETO project used two primary organizational entities. The first was steering committees, which functioned as the organizational meeting place for the key line managers with position power and expertise typically within the areas of IT, R&D, order handling, product management and sales. These managers were often quite familiar with the project due to their involvement in the initial discussions concerning the project. Sometimes also top management representatives named project sponsors attended these meetings. The steering committees thus acted as decision-making bodies in the strategic context of the PSC-project to which the appointed project manager frequently reported back in relations to project progress, budgets, revisions of the project plan etc.

This was found to be important for the general success of the investigated companies as these steering committees signalled to experts as well as users that the CPS-project was to be taken seriously hence a project to which these groups were to dedicate resources. In that respect several project managers emphasize the importance of these steering committees as powerful means of giving the PSC-project status thereby reducing slowdowns often experienced, when expert employees did not find the time to deliver agreed work packages.

However, in almost every case company it was still difficult for these experts to find the time needed, as their assistance was needed in most running and developmental project i.e. they are bottlenecks. And in one smaller company relying heavily on these experts in the running projects, the project team was met by negativity from these experts, because their workload only allowed them to do the work in their spare time. This demonstrates that a high project status can be interpreted as the mere application of management pressure, when the necessary resources are not allocated by the steering committee. In addition these experts also were future users of this particular PCS, and the PCS project was damaged severely.

In putting a guiding coalition together Kotter (1996) speaks of *"reluctant players"* as people to avoid or manage carefully. In this respect one company springs to mind. This company was characterized by a visionary CEO being highly enthusiastic of the PCS technology, but who never managed to create any sense of urgency within the order handling department i.e. the future users of the PCS. Although other factors contributing to the very limited success of the project were present, especially the middle line manager in charge of the order handling department never felt any need for change thereby undermining the credibility of PCS project by not attending project meetings and signalling disinterest to the his staff being the future users of the PCS.

The reason why most likely being that he was afraid of loosing power and influence, as the maintenance of the PCS was to be handled in the quality department. As the order handling department consisted of the most experienced and knowledgeable employees, the middle line manager was highly powerful and due to previous organizational struggles, the CEO did not have the nerve to confront him early in the change process and decided instead to push the PCS project forward without the middle managers approval, support and expertise. Because the unwillingness to confront such managers is quite common, this is also a central reason why change projects fail, as these key employees often block needed action (Kotter, 1996).

Moreover Kotter (1996) advocates the formation of a project team as the second organizational entity with *"the right composition and sufficient trust among members"* especially in times characterized by rapid internal and external changes. Here the necessary time to build this trust is essential, as project team members not committed to the project consciously or unconsciously will have a tendency to signal internal disagreement to experts and future users thereby undermining the

change process. For instance when one company had to replace the sales representative in the project team, the time needed to create and renew a shared understanding of the task at hand was apparently not taken. The sales representative not being as well informed as the rest of the team reacted by not devoting as much energy and commitment to the project and referred during the interview to the PCS project as *"their project"* not *"our project"*.

The project teams are first and foremost in charge of the day to day activities in order to drive the project forward, and the project team members functioned as important points of contact for the rest of the organization to turn to during the different phases of PCS project. Moreover the project team in turn summoned and delegated tasks to a number of co-employees with more specialized expertise these being primarily product experts.

The interdisciplinary nature of PCS-projects was found to depend on the allocation of employees mastering different competencies, the right composition of the project team being primarily expert and user representatives from the areas of product architecture, R&D, selling, prize calculation and PCS development/programming. Although a formal stakeholder analysis as prescribed in the literature was not always carried out, the case studies showed that the above competencies were in fact represented in the majority of the project teams. Thereby *"requisite variety"* (Morgan, 1996) in terms of professional competencies was build into the project team ensuring that the project team was able to match the environment in which the project teams operated.

Moreover there is a tendency that the more successful companies deliberately singled out team members for instance by advertising for project team members internally in the company, which had supplemented their often technical backgrounds with B. Com´s in marketing or organisation. Also project team members including project managers characterized by having extensive cross-functional experience i.e. having worked in different departments within the company or having prior experience from interdisciplinary projects were chosen. Project teams consisting of members having these additional competencies were in accordance with El-Sabaa (2001) found to ease the communication across organizational boundaries, as the project teams knew the terminology of the different knowledge domains at hand.

In this respect several project team members from different companies stated that especially in dealing with experts they felt accepted, as they could meet experts on their turf i.e. go into technically complicated discussions. The combination of technical and business competencies also helped the project teams from loosing sight of the overall strategic perspective and made the individual team members capable of understanding and prioritizing requests and needs for the project team to take into consideration during the development process. This seemed to have had a positive effect on expert and user commitment to the PCS project, as these groups felt assured that the project teams were qualified to weigh inputs and incorporate them into a coherent whole although several compromises had to be made, which did not make allowance for one and every concern.

To our knowledge none of the companies deliberately used the Belbin typology or other concepts for assigning employees to projects dependant on the different roles, which are found important to fill to constitute an optimal team composition (Kousholt, 2001). But focusing on the individual project team roles defined as *"a tendency to behave, contribute and interrelate with others in a particular way"* (Belbin, 2003) during interviews carried out in especially one company, it became clear that several successful teams did in fact consist of team members with different but complementary roles such as; clarifying goals, coordination and administration, team building and networking, analyzing, monitoring and evaluating alternatives, idea creation and creativity, risk identification and management and relieving pressure on other team members by taking on hard and time-consuming operational work assignments.

In a change management perspective none of the twelve companies however formally focused on the team members ability to function as change agents as a criteria for selection. In our view this may have reduced the ability of the project team to establish commitment to and legitimacy of the project in general. For instance in one case the sales representative in the project team did not see it as his role to advocate the PCS project. Thus he never used the extensive information gathering activities within the sales departments to facilitate the understanding of and the integration of the PCS into the every day work environment, which Fowler et al. (1997) defines as being the job of a change agent. Instead he used a recent merger between two departments as the pretext for gathering information not mentioning the PCS at all.

Furthermore the selection of the right project manager is an issue that has met considerably attention in the project management literature. Reviewing a vast amount of project management literature Barber (2004) identifies a body of research dealing with the necessary competencies of successful project managers. A central point is that matching business objectives has recently become more important than matching the technical objectives. Understanding these findings in relations to the PETO project, we find that the more successful project managers have an extensive knowledge of the businesslike and strategic issues as well as a technical understanding and overview of the product structure. In other cases several project managers as well as project team members indicate that their intimate understanding of the relationship between varieties of different knowledge domains as a result of their participation in the PCS project has been valuable in terms of personal learning and career opportunities.

Interdisciplinary competencies in turn made the project managers in cooperation with team members capable of challenging and analyzing expert input and weighing this information against the business scope of the project in the process of making tactical decisions as the project moved ahead. When the project team as a whole was well informed about technical issues as well this was also an advantage in challenging, what Lightfoot (1999) name *"unintentional misrepresentation"*. This notion reflects that experts although well-meaning often still provides information which is incomplete, inconsistent or inaccurate due to forgetfulness, assumptions that further elaborations are not necessary or the difficulties in making tacit knowledge explicit. Although we did not experience the following in the PETO project, this would off course also important, when experts *intentionally* misrepresent their knowledge for instance by withholding information or giving the project team wrong information.

In one company the project manager was an established and well liked figure in which many employees put their trust being sure that he would not advocate a project, which was not in the interest of the other employees. In addition to enjoy very high credibility this project manager also provided leadership throughout the PCS project this being particularly in driving the change process forward (Kotter, 1996). In this respect the project manager was a visionary with a reputation of being involved in or in charge of developmental projects thereby often challenging thinking in grooves.

Communicating the change vision

When communicating the change vision Kotter (1996) recommend the use of many different forums. To obtain further insight into the relative efficacy of different communication channels with respect to influencing user perception with regards of IS implementation; we turn to Agarwal and Prasad (1997). They divide communication channels into a) mass media channels and b) interpersonal communication channels. Adopting this terminology the majority of the cases in the PETO project used group meetings held for the whole sales organization as a mass media channel for providing large amounts of de-personalized information to a wide audience. Here the change vision was communicated and followed up upon as improved functionality could be demonstrated. This also created further awareness in relations to the PCS project.

But other companies supplementing the use of mass customization channels with personal and thus customized communication as the PCS project progressed seem to have had greater success in reducing organizational anxiety concerning layoffs and major changes in the way the employees are used to carry their work in a greater respect than the companies that did not.

For instance in one company the project manager as well as the PCS developer in addition to mass communication frequently used personal communication targeted at individuals typically being experts and future users. As the project manager was a central figure in the company with a large network and impact, many employees frequently relayed on his technical advice and these informal meetings and discussions over lunch showed to be important opportunities to talk about the expected consequences of PCS-project in further detail than allowed at the mass meetings. Thereby misunderstandings were corrected, which otherwise could have led to formation of harmful myths for the project team to handle afterwards. In this respect the project manager and the PCS developer on a regular basis actively formed the adoption decisions of the users, which contributed to the overall success of the PCS project. These findings support the statement of Agarwal and Prasad (1997), who see the creation of awareness through mass media channels as a prerequisite to the development of positive perceptions which, in turn, lead to adoption of the PCS.

In another company the change vision was to obtain growth through gaining additional market shares without having to employ additional employees in the order handling department. This vision was only communicated through mass communication channels these being leaflets and seminars in the early phases of the project. Later on the CEO took for granted that the vision and consequences of the project was well understood throughout the company. But several employees still interpreted his later statement saying that *"several employees working with order handling will not be needed"* to mean that layoffs could be expected although this had never been the intention with the project. This example demonstrates that a heavy reliance on mass communication channels can backfire as the change vision in this case was not fully understood and internalized by the employees. In other words *"seeming inconsistencies"* or mixed signals (Kotter, 1996) were never explicitly addressed. On this basis we suggest that a combination of communication channels throughout the project has to be considered carefully.

Generally speaking every company did in fact emphasize that the PCS project was not about laying people off. Nevertheless employees fearing their jobs were quite common, and it should be mentioned that none of the companies at hand have reached the point where the total economic effects of the PCS are known. This is important as the reduction of the workforce can be rationalized by using other reasons at a later stage thereby concealing possible connections to the PCS projects.

Empowering employees for broad-based action

Kotter (1996) argues that although the first four stages have been taken care of, *"numerous obstacles can still stop employees from creating the needed change"*. In this respect Kotter (1996) state that a common reason why change projects fail is that the guiding coalition does not *"think through carefully enough what new behaviours, skills and attitudes will be needed when changes are initiated"*.

For instance PCS projects call for project teams being capable of learning fast and adapting new techniques and technologies, because the structuring and modelling of the product knowledge quite often calls for specific techniques such as those developed by the Product Modelling Group at the Technical University of Denmark. In addition to master these techniques the project team also have to have educational, communicative and facilitative skills to enable experts to understand and use the very same techniques. Especially since a quite common reaction among experts confronted with the need for the extensive formalisation and structuring of their product knowledge is a high degree of scepticism. For instance when we asked different employees what they thought of the PCS project when first introduced to the idea of product configuration, the most common answer was: *"it can't be done"*. Conclusively learning to master such techniques is one essential example on how employees are empowered for broad-based action.

As mentioned earlier an essential and ongoing task for the project team to take care of is how to handle the employees, who in some way or another will be affected by the development and implementation of the PCS. We shall specifically focus on the future users of the PCS and the experts, which are to deliver knowledge to the project team in order to analyse, and which measures could be or in fact were taken aiming at reducing resistance to change. This is important as resistance or unwillingness to change often arises, when employees are uncertain what the changes imply.

In this respect Agarwal and Prasad (1997) identify three main perceptions, which have been related consistently to predict user adoption i.e. future use of IS these being *relative advantage, ease of use* and *compatibility*. We find that these perceptions represent a further development of Kotters (1996) notions of new behaviour and attitudes, and they will in the following be applied with respect to users as well as experts, as also experts in many cases are involved in the maintenance and further development of the PCS i.e. they are assigned recurrent tasks, which bring them into contact with the PCS on a continually basis as do users. Additionally also experts are characterized by having multiple perceptions towards systems such as PCS.

The notion of *relative advantage* captures the extent to which a user views the PCS as offering advantages over previous ways of performing the same task that is configuring products according to customer specifications. This illustrates the need for the project team to engage in dialogue with users such as sales engineers, sales managers and sales assistants to locate opportunities for designing the PCS in a way that take potential dissatisfaction with existing IT-systems and burdensome work practices into consideration with a view to avoid the reproduction of existing problems to the new work environment and thereby improve the working life of the users.

Although every project team experienced initial scepticism and concerns among users, the successful companies were characterized by an overweight of advantages compared to disadvantages. Also the management of user expectations played a significant role, as successful project teams did not promise more than could be delivered afterwards.

In several companies the PCS replaced outdated and cumbersome IT-systems used for prize calculation and logging customer data and order entries. And in addition the PCS-projects often facilitated a most needed integration of IT-systems, which also improved the working life of the users as updated and correct data now could be found and accessed in one place i.e. the PCS thereby eliminating the cumbersome work in manually transferring data from one system to another.

In contrast one company developed a PCS, which the order handling department could only use for about 20% of the quotes. Thus the employees in the order handling department still had to use the existing system for the remaining tasks. Although tiresome work in terms of typing the same information in several places in fact was circumvented due to the project, the PCS did not provide any relative advantage, this being part of the reason why every internal users to this day only use the old system, as the costs associated with learning the new system and continually switching between systems are viewed as being to high. It should however be mentioned that a number of external users are using the PCS on a frequent basis.

New opportunities for enhancing the job performance of users also emerged. In this respect especially users in engineering companies emphasized the new possibility for using the PCS as a tool for simulating and optimizing different product solutions against variables such as energy consumption, output quality and plant investments. In that way the quality of the sales dialogue with customers was enhanced as the sales engineers felt well prepared before and in between customer meetings. This had not previously been possible as the timeframe in which to respond to customer inquiries had not allowed for this to be done.

Also the frequent delays in the quotation process could be minimized as the sales representatives by use of the PCS did not have to relay on internal experts, which more often than not were too busy to solve or clarify questions right away directed

to them by the sales organization. And another advantage was that the sales representatives no longer had to relay on guesswork in relations to which customer inquiries should be taken more seriously than others, because the PCS made it possible for the sales organization to answer one and all enquiry.

Moreover the PCS were found to constitute foundations for learning and knowledge sharing among the sales representatives as well as experts. The discussions during and after the PCS projects needed for aligning expert knowledge to be embedded in the PCS also proved to be powerful opportunities for knowledge sharing. And sales representatives were empowered to handle each others customers without having to spend lots of time locating the files necessary. And additionally the sales representatives are now affirmed that the product promised to the specific customer in fact can be constructed in the way specified (see also Edwards and Møldrup, 2004).

In this way the project team influenced and shaped the perception of the users by reducing the unwillingness of future users to engage in the always time consuming efforts i.e. personal sacrifices (Kotter, 1996) of unlearning the old way of working and getting acquainted with the new, which always accompany change initiatives. In this respect the relative advantages have compensated for the fact that the PCS never seem to be particularly free of effort to adopt i.e. the *ease of use* have in any case been of a character that requires long lasting efforts for the users.

However, also disadvantages were present. One sales representative operating in the field mentioned that the PCS in many cases is a nuisance, as he now must include office hours in the planning of his daily work in order to configure products instead of doing this manually at home or at hotels. Also a few sales representatives mentioned that they feel inhibited as they to a great extent are forced to sell products within the product range of the PCS. Although this can be sensible in a strategic management perspective forcing the sales force not to sell just the product variants known to the individual sales representative but in fact the whole product range that the company offers the market, this was by quite a few sales representatives seen as constraints in their freedom of action i.e. their work processes. This can also be explained by Coakes and Merchant (1996), which point to that ES quite often are viewed as removing the human element of decision making and individual flair, as the ES is to rigid.

Nevertheless the majority of the project team were capable of establishing and communicating more advantages than disadvantages thereby facilitating commitment and acceptance for the PCS in terms of relative advantages. However, it should be noted that the majority of the PCS projects still are to fulfil these promises, as the PCS projects are still running. The actual consequences in terms of the resulting working life of experts and users are therefore still unknown.

Already we have ventured into the task of managing experts. For a further shading of this central issue in managing and organizing the development of PCS, we now again turn to Lightfoot (1999), who states that *"given the universal desire for power, it is natural for experts to resist any attempt to extract their knowledge and thus weaken their power base… (as) it goes against human nature"*. Given that the power base of these experts is constituted by their expert knowledge and attendant status as trouble-shooters within the organization, the assumption that very few of these experts are willing to give up their knowledge i.e. their power base without getting anything in return seems valid. Therefore Lightfoot (1999) focuses on establishing a classification of experts along with different strategies for motivating these experts with the purpose of making them participate willingly in the development of the ES.

Applying this classification to the case studies done in the PETO project has been an advantage in understanding and explaining the behaviour of experts. Although the PCS project in one case as previously mentioned actually to a large extent was designed to replace experts by PCS technology, the experts did to a high degree participate willingly in the documentation of their knowledge base. The reason why can be explained by the nature of these experts, who closely resembles what Lightfoot (1999) names *"local employees",* who are characterized by their strong loyalty towards the company this being a weighty and rational reason for them to participate in the PCS project, as the retention of their knowledge after their retirement was seen as urgent and essential for the future success of the company. Also these experts were motivated to participate as information provided through the different communication channels convinced them that the PCS project was going to succeed hence it was worth the effort (Lightfoot, 1999).

The majority of the companies also focused on establishing and to some extent communicating relative advantages to the expert groups, as codified expert knowledge in written texts, manuals, technical drawings and the like only to a limited degree were found to compensate for actual expert testimonies and discussions. In this respect a through-going argument was the delegation of routine tasks to the PCS freeing the experts to engage in solving the more complicated tasks also mentioned by Lightfoot (1999), which due to several reasons could or should not be handled by the PCS.

This being an argument that are met with sympathy of the experts, several PCS projects still experienced significant delays, because the experts involvement in the PCS project was seen as an additional task or skeer nuisance (Lightfoot, 1999) to be done whenever a few hours were free from the running projects, which the experts found relatively more important and interesting than participating in PCS projects. For instance expert resistance in one case company was also found to be due to lack of resources to carry out the PCS project alongside the daily work resulting in longer hours than the employees

found reasonable. As one expert stated *"We are not married to the company like the CEO is"*. This is in accordance with Buchanan and Shortliffe (1985) in Lightfoot (1999) identifying this process named knowledge acquisition as *"the critical bottleneck in building ES"*.

Although several companies have chosen to create product configuration teams or indeed whole product configuration departments in charge of maintaining the PCS as well as developing and introducing new PCS, every company rely on the assistance of experts in doing these jobs. Although not up weighing the advantage of being free from repetitive and routine tasks, this is generally found to be a dull task for the experts to engage in. This indicates that Lightfoot (1999) is not always right in assuming that experts feel recognized when assigned maintenance tasks that require their expert knowledge and advice although it could reflect the project teams inability to communicate this to the experts.

All in all this indicate that the predominant expert hesitance to participate willingly in PCS projects can be overcome by establishing relative advantages, as the interviews with experts did not point at resistance due to the redistribution of power as found by Ryan (1988) in Lightfoot (1999). Also the interviews did not reveal any forms of intentional misrepresentation or uncooperative experts refusing to participate (Lightfoot, 1999). But the task of establishing relative advantages for experts still seems more difficult than for users the reason being that ES do not help them directly in doing their job more effectively. This is also reinforced by the fact that other "expert advantages" such as more easily training of experts by means of the PCS target new experts. As such this do not address the experts most needed to cooperate in PCS projects, who in addition might view and interpret the opportunity for training as a threat i.e. that management might use the PCS to find out, which experts are less knowledgeable i.e. most dispensable.

The last perception being *compatibility* describing *"the degree to which the PCS is perceived as being consistent with the existing values, needs and past experiences of potential adopters"* (Rogers, 1983 in Agarwal and Prasad 1997). In this respect one case company circumvented anticipated criticism and resistance to change due to earlier and negative experiences with standardization efforts and IT-projects in general by *"managing the meaning"* of the PCS project (Morgan, 1996). This project team relayed heavily on the term "engineering project" thereby aligning the PCS-project with the existing values within the company and at the same time making it clear that the project unlike past experiences with IT-projects is formulated by engineers having engineers in charge of carrying it through. Thereby making sure that the PCS in the end can be updated and used by engineers without core competencies within the reign of programming i.e. the ease of use is high. In this way the project team carefully manages the meaning of the PCS-project and also signalling that the PCS-project should be assigned the same attention and priority as the running projects within the company.

In yet another company a huge standardization project was terminated a few years earlier than the initiation of the PCS-project, which harmed the early efforts of making the idea lifting off within the organization. When the project team realized the poor experiences with the notion of standardization, the PCS-project was renamed using the concepts of "systematization and modularization" that is concepts that did not conflict with past experiences.

It was also found that especially engineering cultures being used to create a unique product from scratch for every specific customer were characterized by scepticism and disbelief. In this respect the focus on standardization and formalization embedded in PCS projects conflicted with the existing cultures, and experts found the products offered could have been yet more customized or in fact better aesthetically designed than allowed by the PCS.

Furthermore the case studies revealed a tendency that implementation of PCS depends on a transformation from a reactive to a larger focus on proactive behaviour in the organization. Specifying product variations the manual way involved continuous iterations between customers, sales staff and experts entailing employees being used to the demand of quick responses to inquiries in order to serve customers. When PCS were implemented the time pressure remained, but it was now possible for sales staff to reduce the time spent to answer customers dramatically leaving more time for sales representatives and experts to attend assignments and tasks of a developmental nature such as developing new and/or re-designing existing products and processes.

Generating short-term wins

The importance of generating short-term wins has to do with building and sustaining credibility throughout the duration of the project by demonstrating that sacrifices are worthwhile (Kotter, 1996). For instance we have mentioned that a high degree of initial scepticism often were present in the expert communities, which in every company regarded their knowledge base far too complicated to be translated into rules and constraints handled by a PCS. In one company an expert representative in the project team integrated the PCS with an existing simulation tool being utilized to simulate process flows in plants thereby reducing the time spent to do frequent and repetitive tasks from one hour to a few minutes. However in spite of this being a drastic advantage, the expert and member of the PCS project team, did not share this knowledge with his co-employees doing similar tasks. And the example although *"unambiguous and clearly related to the change effort"* (Kotter, 1996) never was used to 1) provide evidence that the sacrifices such as spending time to deliver knowledge to the project team were worthwhile or 2) build momentum in the expert communities thereby turning neutral experts into

supporters and reluctant supporters into active helpers (Kotter, 1996). The short-term win remained invisible to the rest of the organization.

In the same company a knowledge engineer mentioned that having only one development platform in retrospect made it very difficult to demonstrate short-term wins to the future users. The project team did discuss releasing a few subparts of the overarching PCS, which right away had the potential to support sales staff in subtasks. But having only one platform it proved too unstable to handle ongoing development as well as a number of users.

Conclusion

Setting out we claimed that the practice of change management with an emphasis on reducing user and expert resistance would contribute significantly to the overall success when developing and implementing PCS. In that respect the analysis of the twelve case studies individually and in aggregation demonstrated the overall success rate of PCS projects to a large extent can be ascribed the inclusion of considerations and initiatives with a high degree of accordance to change management principles including user participation into the planning and realization of PCS projects.

However, no project manager or project team member represented an explicit and well reflected understanding of change management. Instead especially experienced project managers seemed to have obtained an intuitive understanding of how to practice change management having learned during the years, what actions are sensible and fruitful to take when heading change projects.

But although the PCS projects in the PETO project were characterized as being relatively successful this paper also demonstrates that considerable potential for improvement nevertheless exists. In that respect we believe that early learning about change management including the notion of change agents will be advantageous for project teams taking on PCS projects in order to improve the success rate. Here it is important to remember that a vast number of companies not included in the PETO project have abandoned PCS.

All in all Kotters (1996) change management framework supplemented by research results related to user and expert participation within domains of IS and ES did in our view prove itself useful in order to understand and analyze, which activities and initiatives towards users as well as experts that PCS projects teams must be aware of in order to succeed when developing and implementing PCS. Although the change management perspective is but one of many potentially relevant perspectives such as knowledge management, we especially hope that this article can help practitioners reflect upon the task ahead.

References

Agarwal, R. and Prasad, J. (1997). The antecedents and consequents of user perceptions in information technology adoption. Decision Support Systems 22:15-29.

Alter, S. (1979). Implementation risk analysis. TIMS Studies Manage Sci., vol. 13, no. 2, pp. 103-119.

Anderson, J. and Narasimhan, R. (1979). Assessing implementation risk: A technological approach. Manage Sci., vol. 25, pp. 512-521.

Barber, E. (2004). Benchmarking the management of projects: a review of current thinking. International Journal of Project Management 22:301-307.

Belbin, M. (2003). Management Teams – Why they succeed or fail (2nd edition). Butterworth Heinemann. See also www.belbin.com.

Buchanan, B. G. and Shortliffe, E. H., (eds.) 1985. Rule-based ES, Reading, MA: Addison-Wesley.

Coakes, E. and Merchant, K. (1996). ES: A survey of their use in UK business. Information and Management 30:223-230.

Cummings, T. G. and Worley, C. (2001). Organization Development and Change. Thomson Learning.

DeLone, W. H. and McLean, E. R. (2002). Information Systems Success Revisited. Proceedings of the 35th Hawaii International Conference on Systems Sciences.

Edwards, K. and Møldrup, M. (2004). Unexpected emergence of a Community of Practice when implementing Product Configuration Systems *(forthcoming)*. To be presented at HAAMAHA´2004: Human & Organisational Issues in the Digital Enterprise. 9th International Conference. 25th – 27th August 2004, NUI Galway.

Edwards, K. and Jesper Riis (2004). "Expected and Realized Costs and Benefits when Implementing Product Configuration Systems". Proceedings from the International Design Conference, Design2004, Dubrovnic, May 17th – 20th 2004.

Fowler, P., Martin, G., Juristo, N., Levine, L. and Morant, J. L. (1997). An ES in the Domain of Software Technology Transfer. ES With Applications, Vol. 12, No. 13, pp. 275-300.

Gallagher, S. (1998). Beat the systems management odds. Information Week, vol. 18, pp.61-76.

Jiang, J. J., Klein, G. and Discenza, R. (2001). Information System Succes as Impacted by Risks and Development Strategies.

Kakati, M. (2002). Mass customization – needs to go beyond technology. Humans Systems Management, 21:85-93. IOS Press.

Keegan, A. and Turner, R. (2002). The Management of Innovation in Project-Based Firms. Long Range Planning, vol. 35, pp. 367-388.

Kotter, J. P. (1996). Leading change. Harvard Business School Press.

Kousholt, B. (2001). Projektlederhåndbogen. 1. udgave, 2. oplag. Ingeniøren|bøger, Ingeniøren A/S. In Danish.

Lehaney, B., Warwick, S. and Wisniewski, M (1993). The use of quantitative modelling methods in the UK – some national and international comparisons. Journal of European Business Education 3(2):57-72.

Lightfoot, J. M. (1999). Expert knowledge acquisition and the unwilling expert: a knowledge engineering perspective. ES, August 1999, Vol. 16, No. 3, pp. 141-147.

Morgan, G. (1996). Images of organization. New edition. SAGE publications.

Pine, B. J. et al. (1993). Making mass customization work. Harvard Business Review, September-October: 108-117.

Riis, J., 2003, (in Danish) "Fremgangsmåde for opbygning, implementering og vedligeholdelse af produktmodeller", Ph.D. Thesis, Department of Manufacturing Engineering and Management, Technical University of Denmark.

Rogers, E. M. (1983). Diffusions of innovations. Free Press. New York.

AN ON-LINE INFRASTRUCTURE FOR A
NEW GLOBAL ORGANISATION

A.G.Sheard*, P.Hunter*, K.Orsvärn**

*Fläkt Woods Ltd, Colchester, UK.
**Tacton Systems AB, Stockholm, Sweden.

Email: geoff.sheard@flaktwood.com
Email: paul.hunter@flaktwoods.com
Email: klas.orsvarn@tacton.com

This article describes how a product configuration system may be used to facilitate the creation of a global business infrastructure. Product configuration systems specifically and information management systems generally are shown to underpin post-merger business process re-engineering. They are critical during integration of regional businesses into a single global organisation.

The post-merger integration of three former competitors into a single organisation is described. The key role of product configuration systems in realising synergies within the merged organisation is studied. The post-merger business restructuring is also reviewed. The mechanism by which this transformed the organisation's new product development process is analysed. The article concludes with a consideration of the implications of this technology for managers within the wider organisation.

1. INTRODUCTION

Globalisation is gaining momentum; the shift from a local to a global marketplace is irreversible. As observed by Stewart (2003), the modern global movement started in rural New Hampshire with the 1944 conference held at Bretton Wood. The soon-to-be winners of the Second World War conceived an array of multinational institutions: the World Bank to promote economic development, the International Monetary Fund to support financial stability, the General Agreement on Tariffs and Trade to expand commerce. Stewart (2003) goes on to comment:

> "Their hope was to inoculate the world against economic influenza and rabid nationalism
> that had contributed to the war."

A similar spirit informed the architects of post war Europe, who in 1952 created the European Coal and Steel Community, forbear of today's European Union.   On January 1st, 1995 at a very public ceremony in Geneva, representatives from seventy-six countries signed the charter of the World Trade Organisation. Today China is rapidly emerging as the world's manufacturing base, with Lieberthal & Lieberthal (2003) observing that its importance has occurred only in the past few years as the World Trade Organisation's accession obligations have been implemented.

Kakabadse & Kakabadse (2002) considered the reasons why companies outsource manufacturing of their products. When specifically considering European companies, cost discipline and control were identified as the major motive (Kakabadse & Kakabadse, 2002: 19). This implies a preference for outsourcing in countries with low labour costs to take advantage of the low cost base, but with no interest necessarily in marketing those products within the country of manufacture. The intent is to export back to the west. This perspective was considered by Prahalad & Lieberthal (2003) who concluded that it overlooked the very real opportunity to reach much larger markets further down the socio-economic pyramid. Specifically considering China, a new consumer base is emerging. Starved of choice for over forty years, the rising middle class is hungry for consumer goods and a better quality of life.

Between 1978 and 2002 China's gross domestic product grew by 9.3% annually, three times faster then the American economy did. Multi-national organisations will therefore increasingly see the opportunity for growth in the emerging

economies rather then in traditional home markets. It is, however important not to become myopic about China. As Zeng & Williamson (2003) warn, China is building its own product design capability and developing its own brands. It will soon compete in the west with the very companies who are today re-locating their manufacturing operations there.

This has three primary consequences for European manufacturers. There is a world price for everything – sell at that price or go out of business. Secondly, rising proportions of customers operate globally and expect to be supported globally – do so or be replaced by a competitor that can. Thirdly, traditional European and American markets are generally growing at a fraction of the rate of the Asian economies – to deliver the growth shareholders expect cannot be achieved in traditional home markets alone.

To address the above challenges, mergers and acquisitions unite former competitors, and if the merging organisations are well matched the new organisation has greater combined global reach than any of those acquired. It has global presence. As Selden & Colvin (2003) observed, however, 70% to 80% of mergers and acquisitions fail, meaning they create no wealth for the share owners of the acquiring company. Clearly, the potential synergies between organisations that made merger or acquisition seem so attractive fail to materialise more often then not. This point was considered by Moyer (2004) who observed:

> "Companies come together for many reasons. To achieve economies of scale; to increase profitability; to storm new markets. Yet while there may be total clarity of intent in the design of mergers, acquisitions, and strategic alliances, the outcomes are often disappointing. That is because strategies are ideas; pure with clean lines. Organisations are things; messy and complex. Executives structuring the deals forget that they are uniting two cultures, which is a fancy word for people. Those people see their lives changing. They become stressed and angry. Some leave. Some underperform. Some behave badly. It is also essential to communicate well and calm people's fears as soon as possible. After all, it's scary to look in a mirror and see an economy of scale staring back at you".

The management of an organisation post merger is, therefore, just as important as management of the merger itself. The point was made by Ashkenas & Francis (2000) that there is a need for "integration managers" who are leaders dedicated to the integration process post merger. These integration managers act as electricians who disconnect and reconnect their organisations' wires and as Ashkenas & Francis (2000) observe, "ensure that everyone plays together nicely."

One aspect of integration management, the aspect with which this article is primarily concerned, is the business process re-engineering that embeds the "reconnection" of the merged organisations wires. In this article two key hypothesis are advanced relating to the post merger globalisation of formerly regional organisations:

1. Business process re-engineering is critical post-merger to integrate former competitors into a single global organisation.

2. Being intrinsically independent of geography, the utilisation of a common network underpins the process of creating an integrated global business infrastructure.

The post-merger integration of three former competitors into a new global organisation is described, and the process by which it was created is presented. The critical role of product configuration systems in this process is analysed and the business process re-engineering they enabled considered. The article concludes with a consideration of the implications of this technology for manager within the wider organisation.

1.1 Background

The fan industry has historically been heavily fragmented. No single manufacturer could claim more than a ten percent global market share. Further, each market segment within a specific geographic region has historically been served by different organisations with their own specialist focus. A single manufacturer could command up to an 80% market share in one market segment within one geographic region whilst simultaneously being unable to leverage this position. Manufacturers have historically not been able to take advantage of their channel to market in one geographic region to facilitate entry into different market segments, or the same segment in other geographic regions.

A factor contributing to the above has been the lack of internationally recognised standards within the industry. Over the past forty, and particularly over the last ten years, seven organisations, Table One, have been active, systematically creating a regulatory framework of commercial and industrial fan standards, Table Two. These organisations cooperate, with the global regulatory framework being primarily driven by new and updated International Standards Organisation

(ISO) standards. This emergent regulatory framework has had two primary consequences. Firstly the potential size of the market into which "compliant" products could be sold increased by an order of magnitude. Secondly, the cost of developing products that complied, and were independently tested and certified as proof of compliance increased.

Increasing product development costs have to be recouped. At its most simple, for a fixed manufacturing cost, this must be done either through increased product price, or increased turnover at the same, or lower price. Fierce price competition precludes the former; therefore the only option is increased volume. This is an option made possible by the very standards that had driven the cost of new product development up in the first place. In practice, increase in sales volume per manufacturer has been achieved by consolidation within the industry. In a largely flat global market over the past five years, this has been the only way individual organisations have been able to increase sales volume.

Body	Description
ISO	*International Standards Organisation* Made up of national standards institutes from countries large and small, industrialised, developing or in transition, in all regions of the world. ISO develops voluntary technical standards. These serve as a basis for national standardisation and as references when drafting international tenders and contracts.
IEC	### The International Electrotechnical Commission. The leading global organisation that prepares and publishes international standards for all electrical, electronic and related technologies. This body is the electrical partner to ISO with similar scope, however focused on electrical and electronic devices.
CEN	**The European Committee for Standardisation.** Most standards are prepared at the request of industry. The European Commission can also request the standards bodies to prepare standards in order to implement European legislation. This standardisation is 'mandated' by the Commission, through the Standing Committee of the Directive, in support of the legislation. (Such initiatives are in most cases also supported by the EFTA Secretariat.) The output must therefore be accepted by the Member States, which are delegates to the Committee, and the EFTA countries if they made a similar arrangement. When these standards are prepared in the framework of the 'new approach' directives, they are known as 'harmonised standards' and will be cited in the Official Journal. Products manufactured in accordance with these standards benefit from a 'presumption of conformity' to the essential requirements of a given directive.
CENELEC	**The European Committee for Electro technical Standardisation.** This body is the electrical partner to CEN with similar scope, however focused on electrical and electronic devices.
BSI	**British Standards Institute.** The National Standards Body of the UK, responsible for facilitating, drafting, publishing and marketing British Standards and other guidelines. With collaborative ventures and a strong national and international profile, British Standards is at the heart of the world of standardisation.
EUROVENT	**European Committee of Air Handling and Refrigerating Equipment Manufacturers.** For many years the standardisation was the most important–and sometimes unique–item on the Working

	Groups agenda. Now the work is essentially done in ISO and CEN Committees although for some equipment draft standards are still prepared by Eurovent/Cecomaf Working Groups.
AMCA	**Air Movement Control Association.** AMCA International publishes and distributes standards, references, and application manuals for specifiers, engineers, and others with an interest in air systems to use in the selection, evaluation, and troubleshooting of air system components. Many of AMCA International's standards are accepted as American National Standards.

Table 1. Standards Bodies Regulating the Fan Industry

Number	Title	Status	Comments
ISO 6580	General-purpose industrial fans – circular flanges – dimensions.	Published	First published on 1st September 1981. Under systemic review until June 2006.
ISO 13351	Industrial fans – dimensions.	Published	First published 15th June 1996. Under systemic review till June 2006.
ISO 5801	Industrial fans – performance-testing using standardised airways.	Published	First published 1st June 1997. Now with industry to revise and approve 29th October 2003 draft.
ISO 8428	Industrial fans – influence of the compressibility of the gas being handled.	Technical Report – will not be published.	Decision made April 2003 not to progress this technical report.
ISO 13349	Industrial fans – vocabulary and identification of category.	Published	First published 15th October 1999. Under systemic review until June 2004.
ISO 13350	Industrial fans – performance testing of jet fans.	Published	First published 15th October 1999. Under systemic review until June 2004.
ISO 12499	Industrial fans – mechanical safety of fans – guarding.	Published	First published 15th October 1999. Under systemic review until June 2004.
ISO 5802	Industrial fans – methods of performance testing in-situ.	Published	First published 15th July 2001.
ISO 14694	Industrial fans – specification for balance and vibration.	Published	First published 15th March 2003
ISO 14695	Industrial fans – method of measurement of vibration.	Published	First published 15th April 2003.
ISO 13348	Industrial fans tolerances, method of conversion and technical data presentation.	Committee Draft – not yet published	Committee draft sent to ISO 26th June 2003 for vote.
ISO 13347	Industrial fans – determination of fan sound power level under standardised laboratory conditions.	Draft industry standard. With industry for comment.	Full draft industry standard sent to ISO on the 16th September 2003.
ISO 13347-1	Part 1: General overview.	Draft industry standard. With industry for comment.	Full draft industry standard sent to ISO on the 16th September 2003.
ISO 13347-2	Part 2: Reverberant room method.	Draft industry standard. With industry for comment.	Full draft industry standard sent to ISO on the 16th September 2003.
ISO 13347-3	Part 3: Enveloping surface methods.	Draft industry standard. With industry for comment.	Full draft industry standard sent to ISO on the 16th September 2003.
ISO 13347-4	Part 4: sound intensity method.	Draft industry standard. With industry for comment.	Full draft industry standard sent to ISO on the 16th September 2003.

Table 2. ISO Fan Standards That Have a Direct Impact on the Design and
Use of Fans in Industrial & Commercial Applications

Three European manufacturers have historically held dominant positions in their respective geographic territories and market segments. The Fläkt organisation has traditionally been strong in the Nordic countries, with market share of 70 to 80% in fans, air handling units and other heating, ventilation and air conditioning products. Additionally they competed globally in the power generation, mining and steel market segments. The Woods organisation has traditionally held a similarly dominant position in the UK, additionally selling into the global tunnel ventilation market segment. The Solyvent organisation has held a dominant market position within France for many years, competing globally in the cement market segment.

These three organisations were traditionally competitors, however their respective focus in different geographic regions and on different global market segments resulted in their seldom competing with each other directly. In fact, the most senior engineers within the three organisations were accustomed to working together. Membership of the various standard bodies' technical committees (Table One) that had drafted the very standards driving a need for the merger was drawn from them. As such, the engineering community was at least informally integrated even before the merger occurred.

In February 2001 the carve-out of Woods from GEC and Fläkt and Solyvent from ABB was completed. The three companies were merged into a single legal entity "The Fläkt-Woods Group". The new organisation employed over 3,500 people and had a turnover in excess of 500 million Euros with operations in 29 countries. This made the new organisation the world's largest fan manufacturer, with an overall global market share of approximately 10%.

The new organisation held out the prospect of potential mutual assistance between the merged organisations that offered the possibility of a lower cost base and higher sales volume. Simply consolidating three former competitors into a single organisation does not, however, automatically reduce the new organisations cost base. Similarly, it does not automatically facilitate the "cross selling" of products from one of the merged organisations by the other two. Additionally, the merger in itself did not position the organisation to exploit the market growth in Asia. Although the new organisation had a strong presence in Asian countries and was therefore theoretically better positioned to expand its presence, the process of doing so was not automatic. Realising the potential benefits that had made merger seem so attractive constituted a business challenge that the newly integrated management team had to address, and quickly.

1.2 Business Challenge
The new organisation's business challenge could be summarised as firstly a need for the organisation to become more productive, achieving better business results with less resource, and secondly to become more innovative, creating better products more quickly than competitors.

In his article "Research That Reinvented the Corporation" Brown (1991) suggested that to help senior managers think of innovation in broad terms they should consider that research on new working practices is as important as research on new products. This view is shared by Pearson (1988), who downplays the need for great product or technological breakthroughs, instead encouraging senior management to seek steady small enhancements to all business functions. The subject of innovation was studied by Sheard & Kakabadse (2004) who concluded that working harder alone would not restore lost competitive advantage. Simply focusing on new product development, and working harder to deliver new products faster will not in itself be enough, however well done.

To adequately address the business challenge, the Authors contend that innovation must be considered in terms of process and intellectual property, Figure One. The creation of new intellectual property, in the broadest sense, is the grist of the innovative process. It allows new and better products to be created. This can be characterised as "doing better things" then have previously been done. In contrast the creation of new and innovative business processes may be characterised as "doing things better" then have previously been done. This is the basis of productivity improvement. Truly innovative organisations must do both to innovate more rapidly then competitors, and in so doing restore lost competitive advantage, as over time new innovation becomes state of the art.
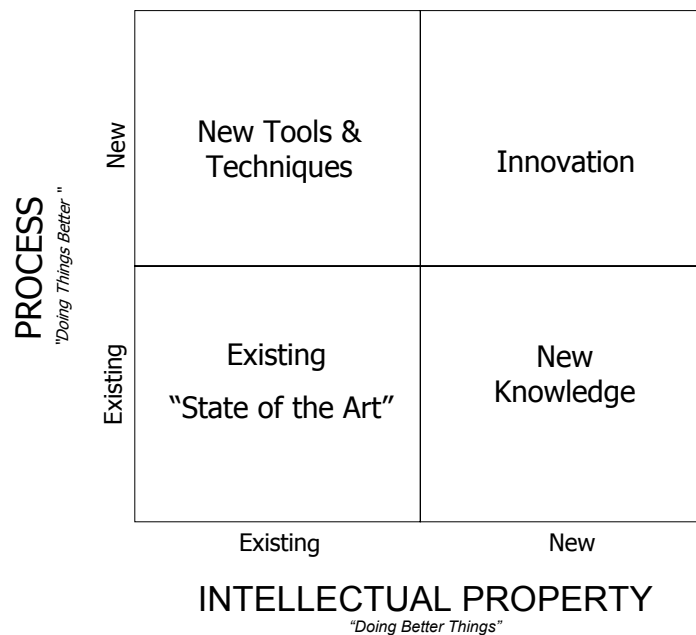

Figure 1. The essence of innovation: doing better things and doing things better.

The process of creating new intellectual property within the context of an engineering organisation may be thought of in terms of the development of new methodologies employing state of the art computational tools, Robinson (2003), Robinson & Norgaard (2003), Teixeira et al (2003) and Corsini et al (2004) or application of state of the art material technology into new products, Sarin et al (2004). These activities can be considered to be part of the classical role of an organisation's research and development team. The process of creating new business processes, however, is not.

Specifically within the context of an engineering organisation, the Authors contend that the engineering team running research and development activity must take responsibility for both "doing better things" and "doing things better". It is the engineering management team's role to drive innovation not only within the classical context of creating new intellectual property, but also within the less familiar context of creating new business processes.

If the above it accepted, then it becomes important that the engineering management team proactively engages in the business of transforming the organisation's business processes. This is true of any engineering organisation, but of particular relevance following a merger or acquisition when the new organisation's business processes are not well aligned and effectively prevent synergies being realised. In this context, business process re-engineering can have a primary impact on the new organisation's productivity, and therefore profitability and ultimately viability.

2. BUISNESS PROCESS RE-ENGINEERING

The challenge posed by integration of the three organisations may be considered from two perspectives. Firstly, they had very different business processes. For example the merged organisation had operations in 29 countries, with a total of 30 different business systems between them. This created a barrier to offering clients "joined up" proposals that included products from across the new organisation. A major reason for the merger was the complementary nature of the products. Collectively it was theoretically possible to offer clients turnkey solutions, both offering the possibility of improving customer satisfaction and increasing total sales.

Secondly, whilst the new organisation had overseas offices, key technical competence was situated in European research and development sites. As the major market growth was in Asia, this proved problematic, as necessary technical data was simply not available on demand in those geographic locations where it was increasingly needed.

The two issues above compounded, as the new organisation had a need to offer turnkey solutions to clients in the growth markets. Only this could ensure that the organisation's potential joint efforts were realised as increasing sales and associated business growth. An inability to compare products from across the new organisation, and to pull that technical data together into a single self-consistent specification and client submittal away from the research and development centers, effectively precluded these synergies being realised.

Although largely complementary, an issue emerged associated with the overlap in product ranges between the merged organisations. There was scope for elimination of some older product ranges and also the benefit of economy of scale associated with selling a larger volume of a rationalised product range. Product rationalization, however proved difficult as the basis on which product performance was calculated and presented in each of the merged organisations was not consistent. It was therefore impossible to undertake a like for like comparison of product performance. This further compounded the two primary issues discussed above, as a single technical submittal needed to be presented to a client in a consistent manner. An inability accurately to compare product performance therefore highlighted a third impediment to realisation of the expected rationalisation.

The new organisation faced a cross-functional and cross-cultural challenge to re-engineer its business processes in such a way as to effectively address the above three issues. Whilst accepting that the challenge was indeed cross-functional, it was recognised soon after the merger that as an engineering global multinational selling heavily engineered products a primary focus of the business process re-engineering must lie within the newly integrated engineering community. The challenge facing the senior management therefore went beyond the obvious issues associated with integrating the three engineering operations.

2.1 Global Reach

A primary requirement for the new organisation was to make technical data about products available in the growing Asian markets. This requirement had two aspects. Firstly, information about all products within the new organisation must be made available to every part of the organisation. If this is not done, local sales operations will continue to focus on sales of those products they are historically familiar with, and not those of the newly enlarged organisation. Secondly, the information is required in a format suitable for distribution. Holding data in local formats within each European research and development centre acts as a barrier to distribution, and therefore sales.

The organisational response to the above was to favour electronic storage of data within Internet enabled database applications. These are intrinsically independent of geography. As such each research and development centre could input product data independently, and critically retain control over its release and issue control. Simultaneously, every sales organisation had access to product data at the moment of release. In this way a common data structure within Internet applications enabled a global network, linking those who created and maintained product data to those who used it.

A secondary requirement for the organisation was to drive cost out of its supply chain. This was addressed both "up" and "down" stream of the organisation. Up-stream, key suppliers were given access to the applications, enabling them to input technical data about components directly, effectively eliminating the need to re-enter data. This helped to reduce human error, and also to ensure component technical data was up to date and accurate. Down-stream, key clients were also given access to the applications. This assisted them in the process of analysing multiple system design concepts using different specification products. In so doing the sales team was relieved of much repetitive re-quoting for the same project, and simultaneously the clients were able to better optimise their system design in a shorter time.

2.2 Front & Back Office

Definition of the new browser based application architecture was initiated via an analysis of the existing front office business systems. A total of four separate and distinct front office business systems were identified, each associated with a different class of product:

1. BTS   Build To Stock
2. ATO   Assemble To Order
3. MTO   Manufactured To Order
4. DTO   Design To Order

The four processes were mapped, Figure Two, time taken to complete each step analysed and error sources identified. These four processes were then streamlined into just two:

1. FC   Fully Configurable
2. CTO   Configured To Order

The first of these new processes encompassed products from the first three of its predecessors, BTS, ATO and MTO. The second new process handles elements of the process MTO whilst adding structure and electronic workflow to the old DTO process. The design of the two new processes formed the initial specification for the new browser based applications. After implementation of the new processes the time taken to generate quotations was reduced by 32%. This was a welcome general administration saving, with reduced opportunity for, and incidence of, human error plus increasing client satisfaction, which may be regarded as a success. The realisation of this saving required a re-organisation within the sales department, transferring some of those who previously prepared quotations to customer facing roles. The new process did not, therefore automatically increase sales department efficiency. They did, however, facilitate a re-structuring within the department that resulted in the average number of quotations produced per day increasing without taking on additional staff.
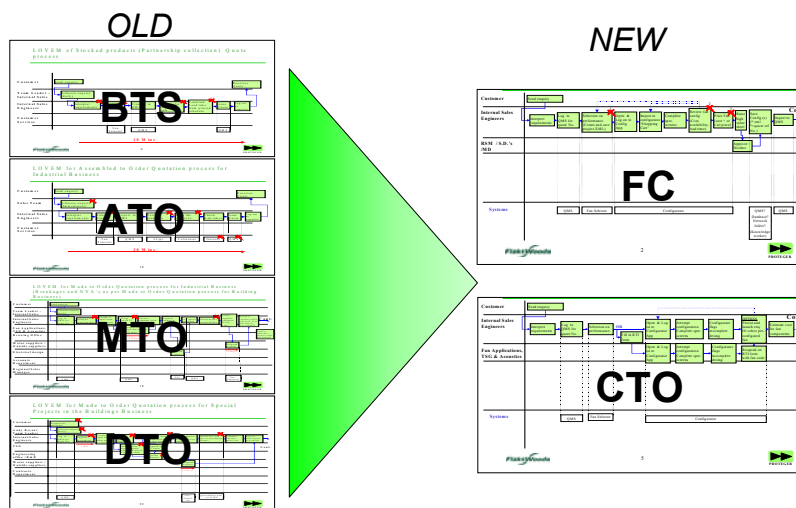

Figure 2. Business process mapping, four business process to two configurator enabled processes.

68

Implementation of the above, however, required more than redesign of the organisations front office processes. Implementation was facilitated by creation and rollout of a radically different new product development process within the engineering function.

2.3 New Product Development

Historically new products within each of the merged organisations were defined within the engineering function, being formally documented by "design" and "performance" reports. These would then be issued to the sales team, who would interpret them when matching client requirements to product specifications. This process was error prone, as the design and performance reports were inevitably at least in part descriptive and therefore implicit in nature. This left scope for interpretation within the front office environment, which invariably resulted in quotations for "standard" products evolving into heavily engineered products after an order had been received.

The descriptive and implicit nature of the new product development process also had implications within the organisation's manufacturing function. This was set up to accept a Bill of Material (BoM) defining every component within a product. A component would either be bought in, or manufactured in house. If a component was to be manufactured in house it would have a drawing number associated with it, where the drawing provides all necessary dimensional information to enable manufacture, plus guidance on the appropriate standards and processes appropriate during manufacture. The manufacturing function was set up and run on the basis of complete and accurate detail about every aspect of a specific product. The design and performance reports, being descriptive and implicit were therefore of little relevance to the manufacturing function.

For all practical purposes, the first that the manufacturing function knew about a new product was when BoMs for it appeared. Supply chains for the bought in components did not exist. Programs for computer-controlled machines were not written. Tooling to facilitate product assembly was not designed, let alone available. As a result, new product introduction tended to be problematic in practice.

The challenge facing the new organisation was compounded by the total separation of the front and back office business processes. Effectively the sales team operated on the assumption that the back office manufacturing function had infinite capacity. Products would be offered on the same standard lead-time, irrespective of the volume sold. As a direct consequence, every time the sales team had a "good" month, the manufacturing function had a "bad" month. Introducing new products into the mix then exacerbated this problem.

The new product development process needed changing to facilitate trouble free introduction of new products. This required information to be made available explicitly to both the sales team and manufacturing function. There was also a business need to link the front office sales team and back office manufacturing functions. Further the local sales teams in Asia needed front office support. Despite the difficulty of addressing these business challenges, the senior management set a goal of addressing all three problems simultaneously. The logic being that the problems were interlinked, and therefore could not be solved sequentially. However challenging, they had to be solved simultaneously if they were to be solved at all.

The new product development process would be changed to facilitate the smooth introduction of new products, and also to link the front and back office plus support the sales team globally. This was the challenge, which could not be met using traditional engineering competencies. This business process re-engineering was recognised as being truly cross functional, with the organisation's information management systems being the critical "glue" that would constitute the links across traditional inter-group boundaries within the new organisation. The business process re-engineering, therefore, would become a series of linked information management projects. The organisation would put its faith in information database technology.

3. INFORMATION MANAGEMENT SYSTEMS

The argument is made by Carr (2003) that there was widespread over investment in information technology in the 1990's. An analogy is drawn between this and investment in railroads in the 1860's. In both cases companies and individuals threw large quantities of money away on half-baked businesses and products. Even worse, the flood of capital led to enormous overcapacity, devastating entire industries.

Let us hope the analogy ends there. The mid-nineteenth century boom in railroads helped produce not only widespread industry overcapacity but also a surge in productivity. This combination set the stage for two solid decades of deflation. Although worldwide economic productivity continued to grow strongly between the mid 1870's and mid 1890's, prices collapsed. In England, the dominant economic power at the time, price levels dropped 40%. In turn business profits evaporated. Companies watched the value of their products erode whilst in the very process of making them.

The world is very different today, of course, and it would be dangerous to assume that history will repeat itself. Companies are, however, struggling to boost profits and the entire world economy is flirting with deflation. It therefore would be dangerous to assume that it cannot.

The above issues are compounded by Austin & Darby (2003) who consider the security implications of our increasingly open and integrated information management systems. When it comes to digital security, there is no such thing as an impenetrable defence, with security breaches affecting 90% of all businesses every year and costing some $17 billion. This is despite the average company spending between 5% and 10% of its information management budget on security.

Given the above, we must consider the prudence of investing in information management systems, making them more open to clients and suppliers as well as across functions and sites within the organisation. The days of open-ended information management budgets are over. Security issues must be of prime concern, as information management systems are conceived, not as bolt on after-thoughts once they are up and running. It is within this context of financial prudence and the vulnerability of information management systems that the implications of business process re-engineering via new and upgraded information management systems must be considered.

Despite the security implications, the authors opted for only internet-based applications as part of the business process re-engineering. The overwhelming advantage of geography-independent solutions must be set against what are ultimately difficult, but addressable security issues. Three browser-based applications were developed, a product selector, product configurator and document archive. When linked into other existing information management systems, Figure Three, the integrated system was able to facilitate transformation of the four historic front office business processes into two processes identified and defined during the business process mapping. This was the process of transforming the new organisation's business processes and creating a browser-based business infrastructure independent of geography.
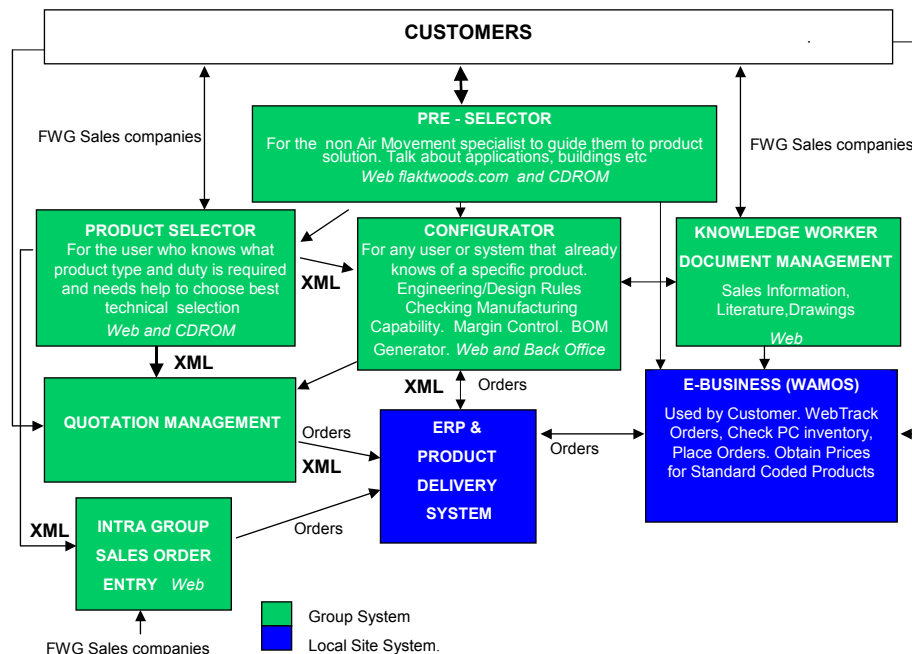
Figure 3. The information management system architecture.

3.1 Product Selector

The first step towards creating a single portfolio of products was identified as the creation of a single product catalogue. The intent of this was initially to make all those within the organisation aware of those products that existed in the parts of the organisation with which they were not personally familiar.

The practicality of creating a single product catalogue proved challenging. As product performance was not calculated in a self-consistent manor in different parts of the organisation, it was necessary to understand the basis on which product performance was calculated and then provide the option to select the preferred method. In effect therefore performance of every product had to be available calculated on the basis favoured by each user.

The above was compounded by a lack of data. Although product catalogues did exist in every part of the organisation that at first appeared complete and comprehensive, the act of gathering them together identified gaps. Scaling and interpolating existing product data could not always fill these. As a direct consequence a substantial performance test program was initiated to generate missing product performance data.

A final technical challenge was associated with the practicality of converting many separate paper based product catalogues into a single browser-based application. This was achieved, Figure Four, using Microsoft technology that facilitated creation of a single browser based product selector. This "on-line" selector was complemented with a full set of product outline drawings that, once a selection was made, could be automatically compiled into a .pdf file with associated product performance data in a form suitable for direct use by clients.
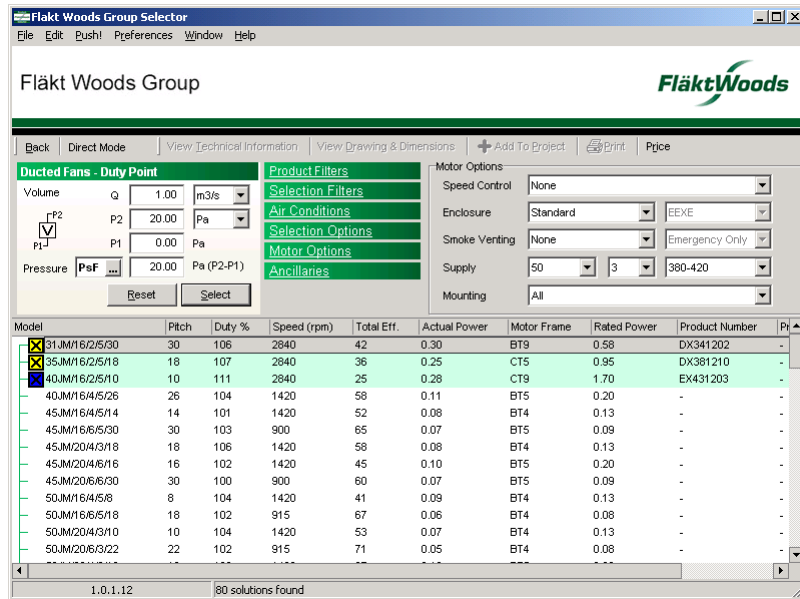
Figure 4. A screen shot from the browser based product selector, effectively an on-line product catalogue enabling identification of products against a specific specification.

The above technical data was complemented by a parallel project aimed at formalising the new organisation's pricing policy. Whilst individual products were relatively easy to cost, the new organisation was faced with over one hundred formally issued price lists for essentially the same products. Price lists were issues in different countries. Within a specific country price would vary depending on the volume a client ordered annually. Some countries employed a network of agents who provided value added services, undertaking some of the work traditionally performed by the manufacturing site.

The process of understanding how the various price lists had been arrived at was a substantial task in itself. Once complete, the factory cost for each specific product was then linked to a set of multipliers to recover overheads, take account of exchange rate risk, volume of product purchased and value added by the local agent. Additionally a language selection capability was incorporated, with the sales teams in each country translating all text into their local language. Once closure was reached on these issues it then became possible to add price, plus commercial terms and conditions to the technical data sheet that could then be created in the language of choice. The on-line product selector therefore evolved into a system that would allow anyone, anywhere, to generate their own quotation. A quotation against which the new organisation would accept an order.

3.2 Configuration
An ability to create a quotation automatically was in itself a major milestone. It made available more of the data required by the sales organisation within Asia to serve clients in a more timely manor without constantly referring back to the European research and development centres. Availability of this system, however, did highlight the seriousness of a known gap in the new organisation's business processes. Specifically, the ability to identify a product and get access to technical data on it provided no real-time information to individual salesmen on when that product might be available or what it would cost and therefore the margin that could be expected, should an order for it be placed.

Information on product lead time, or rather the lack of such information, made clear the separation between front office sales and back office manufacturing processes. The front office system might now be based around an on-line product selector, however the manufacturing functions Enterprise Resource Planning (ERP) system was firmly rooted in a

mainframe computer that generated Bill of Materials (BoMs) and scheduled conversion of the BoM's into finished products totally independently of any promise that the sales team might have made to the client.

This state of affairs had been endemic for years, and was generally accepted as normal, whilst at the same time resulting in significant inter-group conflict within the organisation as those in sales and manufacturing tended, perhaps surprisingly, to actually agree on the root cause of this problem – it was all the engineering function's fault! The individuals within the engineering function took this as evidence that the rest of the organisation was not logical, and therefore it was reasonable to ignore everything they said. Whilst this was helpful in enabling the sales team, manufacturing and engineering function to avoid the need to change anything they were doing, over a thirty year period it had failed to resolve the issue of providing reliable product lead time to the organisation's clients, cost data to salesmen so they could evaluate margin, or indeed to affect any improvement at all in the quality of information passed to clients.

The key to passing data of any sort between the front and back office processes was creation of BoMs. These had historically been generated manually and then embedded in a legacy ERP system within which they subsequently had to be manually maintained. The automatic creation of BoMs on demand was identified as the critical link between front and back office processes. Specifically, the new on-line product selector ultimately generated a "fan code" when a selection was made. This fan code represented a configuration of individual components that collectively comprised the product's BoM. A "configurator" was needed, Figure Five, to generate the BoM from the selected product fan code, interact with the ERP system to establish availability of components and then using knowledge of assembly and shipping time, generate a product lead time and cost. The configuration system therefore had to be knowledge based to accomplish this.
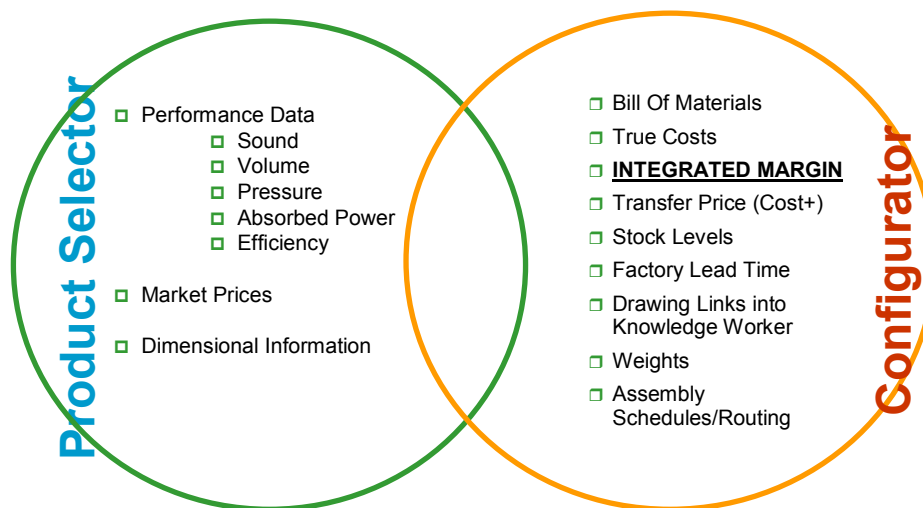
Figure 5. Primary functionality of the on-line product selector and configurator.

For the above to address effectively the issues of linking front and back office systems the configuration had to take place seamlessly as products were selected. It had to be therefore initiated automatically as products were selected on-line, generate a BoM and seek lead-time data on every component within the BoM from the ERP system. Further, should the client choose to place an order, the process of creating a BoM would need to be followed up by actually loading the BoM into the ERP system followed by the generation of an order acknowledgement with a committed delivery date.

3.3 Document Archiving

The loading of a BoM into an ERP system and generation of an order acknowledgement may be regarded as the end of the front office sales process, and simultaneously the start of the manufacturing back office process. In addition to a BoM, primary inputs for the back office manufacturing function are those documents that describe how to manufacture parts, and the process, procedure and standards to which they will be assembled and tested.

The process of generating and storing documents had historically been done using paper based systems. This approach, however, is not compatible with the automatic generation of BoMs and their seamless loading into an ERP system. As BoMs appeared within the manufacturing function it was necessary for all relevant documentation to be made available simultaneously. This issue was addressed by implementing a browser-based document archiving system "Knowledge Worker", Figure Six.
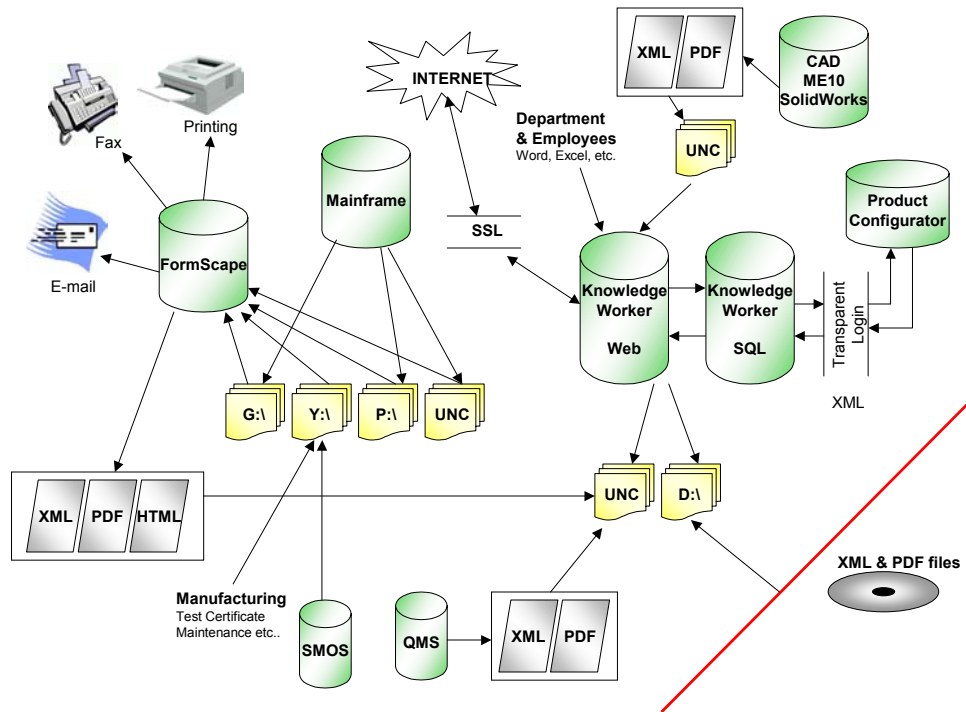
Figure 6. The organisations document management system, based around the
Knowledge Worker browser-based application.

Over a two-year period every product related document was systematically converted into an electronic archival data format. This was helpful to not only the manufacturing function, but also to suppliers who were now able to provide data direct into the organisation's document management system, eliminating much re-keying of data relating to bought in components. It was also useful to clients, particularly those working on large multi-site projects that could use the system as a geography independent repository for project data. Perhaps the single most significant use for the system, however, was as an on-line drawing archive.

3.4 CAD System

The creation of an electronic drawing archive was facilitated by standardisation on the Solid Works three-dimensional (3D) parametric Computer Aided Design (CAD) system. A move to a configurator based methodology for defining new products naturally lends itself to the use of parametric CAD models for defining those products. A parametric model uses a rule-based approach to define a three-dimensional model of a new product, this is intrinsically complementary to the configurator methodology. The use of parametric models also offers the potential to pass attribute files between CAD system and configurator. This facility can be helpful, for example, automating the process of generating customer outline drawings after specific product options that have been chosen.

The above was further enhanced by implementation of the "Smart Team" drawing management system. A drawing management system is the software that generates drawing numbers as drawings are formally released from the engineering function. The drawing management software opens a CAD file, inserts a unique drawing number into the file and then re-saves it. This process was extended such that in addition to re-saving the CAD file with its unique drawing number, the drawing management system also saved a .pdf version of the file. This is an archival format, which is not easily modified and can be used in place of a traditional paper copy of the drawing. This enabled the practice of issuing paper drawings to be eliminated, with the manufacturing function then viewing drawings on screen rather then on paper.

The above was managed within the application "Knowledge Worker". The drawing management software facilitated this via generation of a XML file, containing attributes extracted from the CAD file at the same time the .pdf drawing file was generated. This XML file was then used to populate a drawing database within Knowledge Worker, enabling the .pdf version of a drawing to be located using a Knowledge Worker database search engine. The drawing database was populated with historic data by converting approximately 100,000 two-dimensional CAD files into .pdf files with their attendant XML files. In addition to electronic storage of drawings, Knowledge Worker managed the version control of drawings. Old versions were retained, however without administrator rights a user could only access the most recent version of each drawing. This facility made a useful general administration saving, as it was no longer necessary manually to maintain and copy drawings.

With the organisation's drawing archive now held in a Knowledge Worker database, the process of accessing drawings associated with part numbers within a BoM was automated. The new CAD system and drawing management system therefore support the configurator process in two ways. Firstly, the rule-based process of creating parametric models is complementary to the knowledge-based process of defining products with a configurator. Secondly, once products are defined, the issue of drawings into Knowledge Worker enables them to be accessed and retrieved automatically within the manufacturing function specifically and the wider organisation generally.

4. CONFIGURED PRODUCT

The definition of "product configuration" was considered by Mittal (1989) who defined a configurable product line as occurring when customisation is combined with rational production and handling. This is achieved by using a set of pre-defined components that can be combined in different ways to satisfy a range of requirements. In a product configuration task, the input is a requirements specification, and the output a quantified list of components in a configurable product line. The components together must satisfy the input requirements and adhere to the technical restrictions of the product line. The output is called a configuration.

It is usually the case that the same functional requirements can be satisfied with different combinations of components, and different configurations are possible. In this case, the output configuration should be sufficiently optimal, implicitly or explicitly, according to some criterion, which is usually some kind of cost.

The above definition of product configuration is very wide. Orsvärn & Axling (1999) further specified the definition of configuration, concluding that inherent in the notion of a configurable product line is not only its pre-defined set of components, but also a modular breakdown structure. The breakdown structure is a "part-of" structure of abstract configurable parts. The levels of the part-of structure are components. In contrast to a component, a configurable part can satisfy a wide range of varying requirements. The requirements input to a configuration task, is the requirements on a configurable part.

Orsvärn & Axling (1999) argued that a configuration could be regarded as an instantiation of a configurable part. In this context instantiation is defined as the process of deriving an individual statement from a general one by replacing the variable with a referring expression. A configurable part can be instantiated in many different ways, to satisfy different requirements. The modular breakdown structure, however, is a property that makes configuration routine. A second important property of a configurable product line is that the requirements it can satisfy can be represented with a pre-defined set of parameters, where each parameter has a pre-defined set of possible values. The set of possible values of a parameter may be infinite in principle, but in practice it is finite. A parameter may represent an attribute of the configuration as a whole, or an attribute of a part of the configuration, such as a key component selection.

The specific requirements required to satisfy the definition of "configuration" as defined by Orsvärn & Axling (1999) is an assignment of values to parameters, and more generally restrictions on parameter values. The configuration requirements may refer to several configurable parts, and several instances of those configurable parts, if these have interdependencies.

A configurator engine is a computer program that performs the configuration task, matching input requirements with a suitable configuration, interactively assisting the user in finding a sufficiently optimal configuration for the customer's needs. It uses a representation of relevant knowledge of the configurable product line, called a configuration model.

The Tacton Configurator, Orsvarn & Axling (1999), adheres to the above principles and is based on research and application at the Swedish Institute of Computer Science, Axling & Haridi (1996) and Orsvarn & Hansson (1996). By using a descriptive constraint based representation of the product knowledge, in a graphical modelling environment, Figure Seven, engineers can maintain the configurator without programming skills.

In addition to the configuration capabilities, an important configurator requirement for Fläkt Woods was to introduce it as quickly as possible, without unnecessary changes to existing legacy applications. The product selector already existed and was undergoing a web enabling and extension to the whole fan product scope of the organisation. The different manufacturing units had different back-end systems.

The Tacton Configurator is designed to extend other applications, to enable them to handle configurable products. It is able to directly load relevant product data live from any data source, which means that the configurator can use the same product database as the product selector, while applying cost and lead-time data directly from the different back-end

systems. Moreover, the configurator also produces a highly customizable XML output that can contain information needed to drive the different back-end systems, as well as quotations and parametric CAD data.
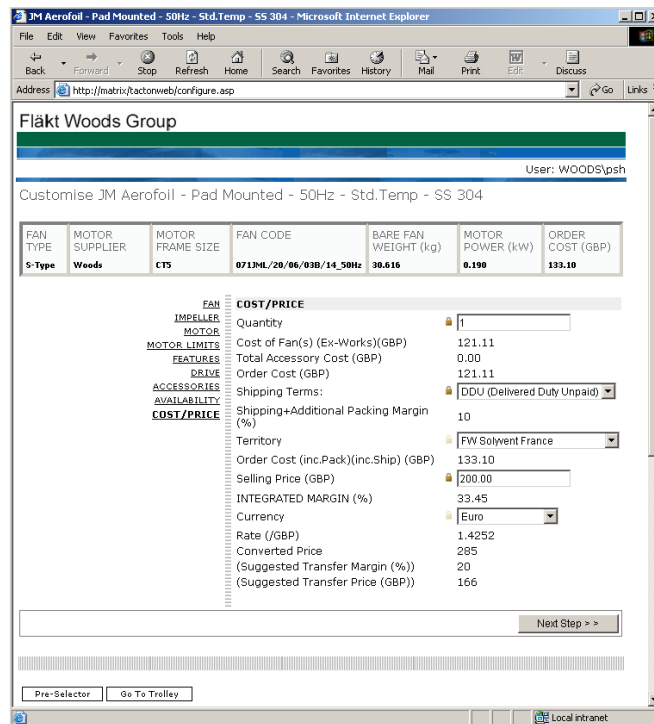
Figure 7. A screen shot from the Tacton configurator.

4.1 Change Management

The product configurator was different to other information management systems implemented as the realisation of the benefit associated with it required significant organizational change within the sales department. It could be argued, for example, that the browser-based product selection tool is no different, from a process perspective, to a paper catalogue. Similarly an electronic drawing and document archive could be regarded as no different, again from a process perspective, to a paper archive. These information management systems constitute change from the point of view of those using them, however the nature of the change is non-intrusive. That is to say, what people do remains the same, however the tools they have available to help them are simply brought up to date by "going electronic".

The browser-based drawing archive, for example, facilitated automated version control of drawings – the correct version was always issued with a BoM. As such the endemic problem within the manufacturing function of re-manufacturing parts after initially making them to the wrong issue of a drawings was effectively addressed, ensuring ready acceptance of the new system within the manufacturing function. The organisational change was, therefore not felt within the manufacturing function. The fact that they were now linked to the sales department via the new browser-based tools was a matter of supreme indifference to those in both the sales department and manufacturing function.

In contrast to the manufacturing function, the product configurator did require change to established working practices within the sales department. The sales team had historically been organized around those who were "customer facing" and those who provided "backup", preparing quotations and resolving technical issues at the request of those who were customer facing. Implementation of the product configurator reduced the need for backup, with the sales team promptly adapting the way they worked to include additional work previously not done because of a lack of time prior to submitting quotations. This effectively reduced the technical and commercial risk associated with making a quotation, but had no impact on the average number of quotations produced per day.

The productivity improvement was traded for reduced technical and commercial risk. Realising the productivity improvement required some of those who had traditionally provided backup now becoming customer facing. This was deeply unpopular amongst those forced to change. Those continuing to provide backup, who perceived their workload to have increased, also resented it. Further, it was resented by those who had previously been customer facing, who perceived

their backup to have reduced and as such increasing the workload on themselves. As such realising the benefits of implementing the product configurator within the sales department required significant change in working practices that in turn required a focused and sustained management effort to embed.

The product configurator may have been unpopular within the sales department, however across the wider organization it was invisible. Lead-time information was now available within the sales team, however it was via the now familiar electronic product selector. The fact that a configurator had been involved, interacting with the organisations ERP system, Figure Eight, was of no interest what so ever to the individual salesmen and clients using the product selector. Similarly, the manufacturing function had always received BoMs from the ERP system. The fact that they were now created via a product configurator was a matter of such indifference within the manufacturing function that when the switch from legacy system to product configurator generated BoMs was made, no one within manufacturing noticed.
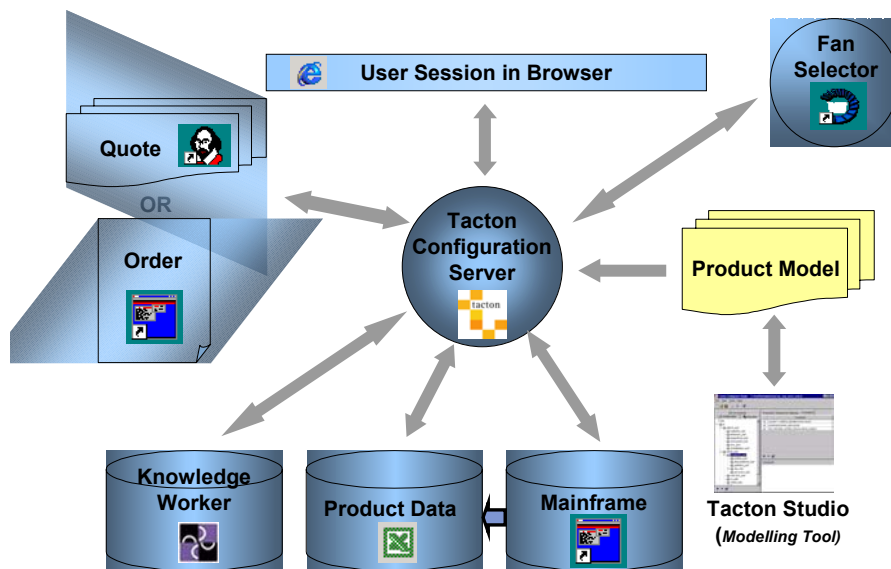
Figure 8. Tacton configurator interfaces.

The above should not be taken as evidence that the product configuration system was implemented without effort. Seamless integration between product selector and ERP system was achieved by linking the product selector to the configurator, and the configurator to the ERP system via a bi-directional piece of middleware. This middleware comprised an information management project in its own right, and represented a major challenge. It was, however, invisible to the wider organisation.

4.2 Product Definition

The organisational change was felt within the sales department, however the most fundamental change to working practices took place within the engineering function. It was here that the new product development process was implemented.

The historic performance and design reports were now obsolete. The product selector required performance data to be issued into its database. The configurator required design data to be defined as configurator models. The shift to parametric 3D CAD models of new products, with a knowledge-based definition of that product being then built within the configurator was radically different to the historic approach to new product development. Historically products would be defined using a 2D CAD system that was little more then an electronic drawing board. The move to parametric 3D CAD enabled rapid prototypes of parts to be produced that whilst unsuitable for use in production products were quite adequate for use in prototype or pre-production products. These rapid prototypes enabled product performance to be established before investing in expensive tooling, enabling an iterative approach to be taken and facilitating optimisation of product performance.

The most fundamental change in the new product development process was, however, linked to the knowledge-based definition of a product within a configurator model. The historic practice of writing design reports, whilst rigorous, did not compare to the level of explicit detail required to build a configurator model. As such the time and effort required to build configurator models was high when compared to that historically taken to create a design report. The opportunity for error within the front office when interpreting the result, however, was drastically reduced.

A consequence of a move to configured products was a step change in the detail with which new products were defined. Essentially the new product development process now demanded much more time to be invested in product definition. The engineering function had habitually complained about the level of order-related engineering and the inability of the sales team to sell standard products. This perspective denied the reality that some issues were overlooked in traditional design reports, and secondly that their implicit nature made interpretation within the front office difficult. In short, the sales team's inability to sell standard products was at least in part a consequence of the engineering team's inability to define them. By characterising the sales team negatively, the engineering function avoided the reality that it needed to do better.

To simply say that any part of an organisation "must do better" is, however, unhelpful. Almost without exception people within organisations do the best they can with what they have. It is a role of senior management to identify business infrastructure issues that constrain organisational performance and address them. Any senior manager who habitually criticises individuals and departments within an organisation as needing to do better is often avoiding the issue that it is in fact them, not those they criticise, who must do better. Introduction of a configurator-based approach to new product design drove product definition to a new level of detail and addressed a root cause of inter-group conflict within the wider organisation. The new product development process had a positive impact not only within the engineering function, but also within the wider organisation.

The positive aspects associated with a move to configured products were associated with a new requirement to ensure that the configurator models always remained a true representation of current operational capabilities. There was a need to maintain and update existing configurator models, as well as create new ones. The work necessary to do this was split between two steering groups. The Configuration Steering Group (CST) made strategic decisions on which products new configurator models should be created. The Configurator Authorisation Team (CTA) made tactical decisions of how existing models should be improved and upgraded. Membership of the CST and CAT comprised representatives from different functions within the organization, Figure Nine, to ensure that any decision was made cross-functionally.
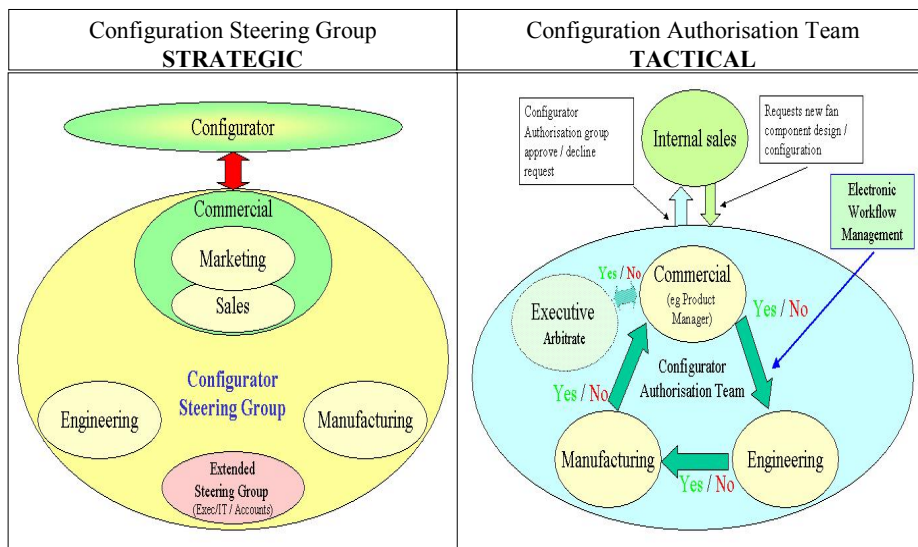

Figure 9. Cross-functional configurator management teams, set up to manage the decision making process associated with prioritisation of new product models, and maintenance of existing models.

4.3 A New Infrastructure

The new organisational infrastructure facilitated smoother operation of the organisation. An example of this is the increase in tunnel and metro projects with a value of over £1,000,000. Historically the merged organisations would collectively have worked on no more than two at any one time. The new organisation was working on ten within a year of the merger, whilst simultaneously reducing the total number of people employed. The organisation was doing more with less.

A move to an on-line browser based infrastructure reduced the importance of geography to an extent that facilitated effective support of local product manufacture. It is common commercial practice in some countries to mandate a minimum local content in any large project, therefore this support can be critical when winning and subsequently delivering large projects.

The provision of cost data in a more accessible form enabled greater transparency when bidding to clients which in turn translated into better control of both product and project "integrated margin" across the organisation. This helped to ensure orders were taken at more predictable margin. Additionally, it enabled the organisation to reject those projects on which a loss was predicted, and so helped to avoid those projects that would otherwise have become loss making. Overall, the economic consequences of the new on-line browser based infrastructure were positive. Administrative costs reduced, margin control improved and business growth in Asia particularly was supported.

5. CONCLUSIONS

The issues surrounding information technology enabled communication and organisational effectiveness were studied by Kakabadse et al (2000) who observe that in the organisational setting, communication is the mutual exchange of meaning between active participants. Complex organisations consist of many social and cultural groupings and communication between them is likely to involve not only shared meaning but also contradictory and contested meanings, thus requiring value and conflict resolution. The usefulness of new technology is illustrated by Kakabadse et al (2000) by citing Plato's (1997) *The Phaedara* where the usefulness of a new technology, the written word, was explained to the Pharaoh, by its creator, Theuth, the Egyptian god of invention, in the following manner:

> ".… this invention, O King, will make the Egyptians wiser and will improve their memories; for it is an elixir of wisdom and memory that I have discovered'. However, Thomas, the Pharaoh, replied: 'Most ingenious Theuth, one man has the ability to beget arts, but to judge of their usefulness or harmfulness to their users belongs to another; and now you who are the father of letters may have been led by your affection to ascribe to them a power the opposite of that which they really possess. For this invention will produce forgetfulness in the minds of those who learn to use it, because they will not practice their memory'.

The Pharaoh was correct in his view that this new form of communication would result in a new social situation, the consequence of which was not necessarily universally beneficial. The new forms of communication created by information technology were studied by Sproull and Kiesler (1991) who postulate that the they also create a new social situation that is impoverished in social cues and shared experience as electronically mediated messages are likely to show less social awareness. This occurs at the very time that Kakadabse et al (2000) make the case for improved communication to facilitate increased value and conflict resolution between the social and cultural groupings who now find themselves seamlessly connected within the complex organisation. It is, therefore, entirely possible that new and improved information technology systems will both make data available around a complex organisation and simultaneously promote inter-group rivalry within it ensuring that no value is derived from provision of the data.

This issue is not an information management issue in itself; it is a wider problem within a complex organisation exacerbated by the lack of social cues associated with electronic communication. The observation of Wick and Leon (1995) may help put the issue into context: in learning organisations, information flows with speed and honesty between all organisation actors. Openness is the lynchpin of an organisation's ability to scan and position itself in the market and political space. By equalising the power between leader and followers, leaders are often able to gather information about what really is going on instead of what they hope might be going on in organisations. Learning organisations structure themselves and utilise information technology in a way that speeds the flow of internal information. They create cultural norms that place a high value on honesty, even in the face of difficulties. Instead of covering up problems, learning organisations make problems visible in order to encourage participation in finding solutions quickly. This is an issue of leadership and the role of the senior management within a complex organisation. The information management systems that constitute the organisational infrastructure cannot in themselves alone provide the leadership an organisation needs. They are, however essential tools for those senior managers who presume to step into leadership.

6. REFERENCES

25. Ashkenas, R.N. and Francis, S.C. (2000). Integration Managers: Special Leaders for Special Times. Harvard Business Review, November – December.

26. Austin, R.D. & Darby, A.R. (2003). The Myth of Secure Computing. Harvard Business Review R0306J, June, pp 120-126.

27. Axling, T. & Haridi, S. (1996). A Tool for Developing Interactive Configuration Applications. Journal of Logic Programming, Vol. 26, Issue 2.

28. Brown, J.S. (1991). Research That Reinvents the Corporation. Harvard Business Review No. R0208G.

29. Carr, N.G. (2003). IT Doesn't Matter. Harvard Business Review R0305B, May, pp 41-49.

30. Corsini, A., Franco, R., Sheard, A. G. & Kinghorn, I. R. (2004). The Aerodynamic Interaction of Tip Leakage and Mainstream Flows in a Fully-Ducted Axial Fan, ASME Paper No. GT2004-53408.

31. Kakabadse, N.K., Kouzmin, A. and Kakabadse, A.K. (2000). Information Technology – Enabled Communication and Organisational Effectiveness. In Creating Futures: Leading Change Through Information Systems (Ed.: N.K.Kakabadse and A.K.Kakabadse). Ashgate Publishing Ltd., Aldershot, pp 101-125.

32. Kakabadse, A.P. & Kakabadse, N. (2002). Smart Sourcing: International Best Practice. Palgrave, New York.

33. Lieberthal, K. & Lieberthal, G. (2003). The Great Transition. Harvard Business Review R0310D, October, pp 70-81.

34. Longet, C. & Battisoni, F. (2003). Design of Low Noise Fan. Proceedings of the 2nd International CETIAT and CETIM Symposium on Fan Noise, Senlis, France, 23-25 September.

35. Mittal, S. & Frayman, F. (1989). Towards a Generic Model of Configuration Tasks. Proceedings of the 11th IJCAI.

36. Moyer, D. (2004). The Sin in Synergy. Harvard Business Review, March, pp 131.

37. Pearson, A. (1988). Tough-Minded Ways to Get Innovative. Harvard Business Review No. R0208H.

38. Plato (1997). The Phaedara. In Plato: The Complete Works (Ed.: J.M.Cooper and D.S.Hutchinson). Hacket Publishing Company, New York, pp 230-238.

39. Prahalad, C.K. & Lieberthal K. (2003). The End of Corporate Imperialism. Harvard Business Review R0308G, August, pp 109-117.

40. Sheard, A.G. & Kakabadse A P (2004). A Process Perspective on Leadership & Team Development. Journal of Management Development, Vol. 23 Issue 1; Monograph Issue, February pp 6-106.

41. Sproull, L. and Kiesler, S. (1991). A Two-Level Perspective on Electronic Mail in Organisations. Journal of Organisational Computing, Vol 2, No 1, pp 125-134.

42. Stewart, T.A. (2003). Mending Walls? Harvard Business Review, August, pp 8.

43. Orsvärn, K. and Hansson, O. (1996). Pre-study of Configuration of a Naval Combat Management System. In Working Notes of AAAI Fall Symposium on Configuration, MIT, Nov. 1996.

44. Orsvärn, K. and Axling, T. (1999). The Tacton View of Configuration Tasks and Engines. In Configuration – Papers of the AAAI'99 Workshop. Technical Report WS-99-05, AAAI Press.

45. Robinson, C. J. (2003). Application of CFD to the Design and Analysis of Axial and Centrifugal Fans and Compressors. In Advances of CFD in Fluid Machinery Design (Ed.: R.L.Elder, A.Tourlidakis and M.K.Yates). Professional Engineering Publishing, London, pp 53-64.

46. Robinson, C. J. & Norgaard, C. M. (2003). Design of a Ducted Axial Flow Fan. Proceedings of the Seminar Advances of CFD in Fluid Machinery Design, Institution of Mechanical Engineers, London, November 18th.

47. Sarin, D. A., Majamaki, O. A., Kuokkanen, J. K. & Sheard, A. G. (2004). Centrifugal Compressor Redesign Utilising Carbon Fibre Composite Impellers. Proceedings of the 2nd Seminar on Advanced Materials for Fluid Machinery, Institution of Mechanical Engineers, London, February 26th.

48. Selden, L. and Colvin, G. (2003). M&A Needn't be a Loser's Game. Harvard Business Review R0306D, June, pp 70-79.

49. Wick, C.W. and Leon, L.S. (1995). From Ideas to Action: Creating a Learning Organisation. Human Resource Management, Vol. 34, No 2, Summer, pp 299-311.

50. Zeng, M. and Williamson, P.J. (2003). The Hidden Dragons. Harvard Business Review R0310F, October, pp 92-99.

Mass Customization for EVOLVING Product Families

Thorsten Krebs, Katharina Wolter

Laboratory for Artificial Intelligence
University of Hamburg
Vogt-Köln-Str. 30
Germany, Hamburg, 22527
Thorsten Krebs, krebs@informatik.uni-hamburg.de
Katharina Wolter, kwolter@informatik.uni-hamburg.de

Lothar Hotz

HITeC c/o
University of Hamburg
Vogt-Köln-Str. 30
Germany, Hamburg, 22527
Lothar Hotz, hotz@informatik.uni-hamburg.de

Abstract: Evolution of products is inevitable throughout their life cycle – driven by advancing technology, increasing customer requirements or bug fixes. Therefore, the set of components as well as the dependencies between those components are getting more complex and only few experts are able to configure products. But in mass customization scenarios it is desired to generate products specific to customer requirements. One promising approach to this is abstracting from the available components and their interdependencies by focusing on features. In an extended configuration process, the customer can select a set of features for the desired product and based on pre-defined mappings the system architecture is selected and the corresponding product components are inferred. This approach hides modifications to product components by showing the same or only slightly changed features to the customer.

1. INTRODUCTION

Product configurators are widely used to generate customer-specific products out of a vast amount of potential variants. This combines the two essentials of mass customization: adapting the derivation process by producing standardized assets and assembling these assets to make the unique product for the customer.

Evolution of products and product components is inevitable throughout their life cycle – driven by advancing technology, increasing customer requirements or bug fixes. Therefore, the set of components as well as the dependencies between those components are getting more complex and only few experts are able to configure products based on such complex configuration models. But in mass customization scenarios it is desired to generate products specific to customer requirements. Thus, the configuration process would largely benefit when everybody is able to configure products – e.g. in internet configuration scenarios (compare [Ardissono et al. 2002]).

One promising approach to this is abstracting from technical details of the available components and their interdependencies by focusing on functionality. Functionality of products and product components can be represented with features (see also [Kang et al. 1990]). In general, the expected improvements of using a feature-based approach for software development are better control over variability, extended reuse of requirements, improved configuration support and sales support as well as improvements on new development [Hein et al. 2000].

The structure-based configuration method [Günter 1995] can be used to support the feature-based approach. In such a feature-based configuration process the customer can select a set of features for the desired product and based on pre-defined mappings between features and the product line artifacts a (potentially automated) process constructs the system

architecture by selecting the corresponding product components (hidden for the user). This approach gives the possibility of hiding modifications to product components by showing the same or only slightly changed features to the customer.

The remainder of this paper is structured as follows: in Sections 2 we show how features and their interrelations can be described in detail. Next, we explain how features, software artifacts, hardware artifacts and the mapping between them can be defined in the configuration model (Section 3). How this configuration model is used to derive complex products is explained in Section 4. Finally, the aspect of evolving product components is introduced and change operations used to capture the modification of configuration models are described (Section 5). A short outlook concludes the paper.

2. FEATURES

We follow the definition in [Kang et al. 1990] where a feature is defined as "a prominent or distinctive user-visible aspect, quality or characteristic of a software system or systems". Further, "a feature model is a specific domain model covering the features and their relationships within a domain" [Ferber et al. 2002]. Features are modeled hierarchically in an ontology that can consist of taxonomy and partonomy. Taxonomic relations (i.e. specializations) allow defining more specific system properties and therefore a distinction between different types of features. Compositional relations provide the means for grouping related features by placing them next to each other (subfeature relation). Feature hierarchies typically contain facilities to express diverse types of variability:

- *Mandatory* features are present in all products that belong to the product family in the modeled domain.
- *Optional* features may or may not be included in a product. If an optional feature is not part of the product, all subfeatures of that feature are also excluded.
- *Alternative* features represent a choice between a set of features from which exactly one has to be chosen.
- *Multiple* features capture the possibility to choose multiple features from a set of features, but at least one has to be chosen.

Properties of product lines are common to all product line members – i.e. they are mandatory features. Mandatory features represent basic functionality of all products in the given domain. Optional features are not included in all but only in some selected products and represent the admissible differences between product line members.

Because features can be interconnected not only by hierarchical relations (i.e. taxonomies and partonomies) but also by dependencies concerning arbitrary features, more sophisticated variability mechanisms are needed. The following is a list of further needed dependencies (compare [Brüne et al. 2003], [Krebs and Hotz 2003]):

- *Requires:* Features can be required by other features - i.e. the existence of the required feature is needed for the former one.
- *Excludes:* Features may exclude each other. This happens when two features can not be selected together, e.g. when the system components that realize these features are incompatible. This is a mutual exclusion.
- *Recommends:* A weak form of the requires relation is a recommendation. The existence of features can be recommended for other features. This can also be seen as the semantics of a default value.
- *Discourages:* Contrary to the latter, features may be discouraged for other features in the system. This is a weak form of the mutual exclusion. Hence, it describes that a feature is not chosen per default.

Due to views of various granularities, features are utilized to model different levels of abstraction. [Kang et al. 2002] e.g. distinguish between product *capabilities*, the *operating environment*, *domain technologies* and *implementation techniques*. While product capabilities are general terms that also customers can understand and select the desired functionality from, implementation techniques are usually hidden from customers and used by application engineers that implement products or product artifacts. Therefore, mappings between features on the diverse levels can be modeled through taxonomic and compositional relations and dependencies.

3. MODELING WITH STRUCTURE-BASED CONFIGURATION

In product families besides features also artifacts (i.e. software modules and hardware components) are considered. In our approach features, artifacts and the mapping between these are formalized in a *configuration model* based on mechanisms from structure-based configuration. This model is further used for automatic product configuration. Features, software modules and hardware components can be represented with *concepts* provided by the modeling facilities of structure-based configuration. A definition is given in the following:

- Each concept has a *name* which represents a uniquely identifiable character string.
- Concepts are related to exactly one other concept in the taxonomic relation. Thus, the latter concept is the *superconcept* of the former one.
- Attributes of concepts can be represented through *parameters*. A parameter is a tuple consisting of a name and a value descriptor. Diverse types of domains are predefined for value descriptors – e.g. integers, floats, sets and ranges.

- Partonomies are generated by modeling *compositional (has-parts) relations*. Such a relation definition contains a list of parts (i.e. other concept definitions) that are identified by their names. Each of these parts is assigned with a minimum and a maximum cardinality that together specify how many instances of these concepts can be instantiated as parts of the aggregate.

   Has-parts relations are a class of compositional relations. Thus, the knowledge engineer is free to define his own relations and give his own names – e.g. a concept can contain a has-features definition next to a has-parts definition to emphasize on the difference between artifacts and features.

```
Concept
 name: Software Module
 superconcept: Software Artifact
 parameters:
  Size [1 1024]
 relations:
  part-of
   System
```
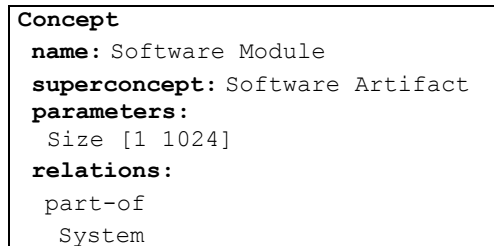
Figure 1. Concept Definition

Concepts can be utilized to model all kinds of entities of the product family like software and hardware artifacts or features. Through the taxonomic relation these entities are grouped next to each other in one configuration model – each having their own branch in the hierarchy. Therefore, we have defined a *common applicable model (CAM)* which contains predefined concepts with properties common to all these entities. An earlier version – called *upper model* – was already introduced in [Hotz and Krebs 2003]. In Figure 1 the definition of the concept *software module* is depicted. The entry *superconcept: software artifact* indicates that the software module is taxonomically placed under the concept software artifact which is predefined in the common applicable model. One parameter is defined: the *size* of the software that can range between one kilobyte and one megabyte. Further an inverse definition of the compositional relation (i.e. *part-of*) is given to express that this *software module* belongs to a *system*.

Figure 2 shows the CAM that contains definitions for the basic entities *feature*, *hardware artifact*, *software artifact* and the *system* that represents configurable product family members and contains the configurable artifacts through compositional relations. Application-specific concepts are introduced into the configuration model by simply placing them under the given entities – e.g. a *software module* is introduced by taxonomically relating it to the concept definition of the *software artifact* (*software artifact* is superconcept of the new *software module*).
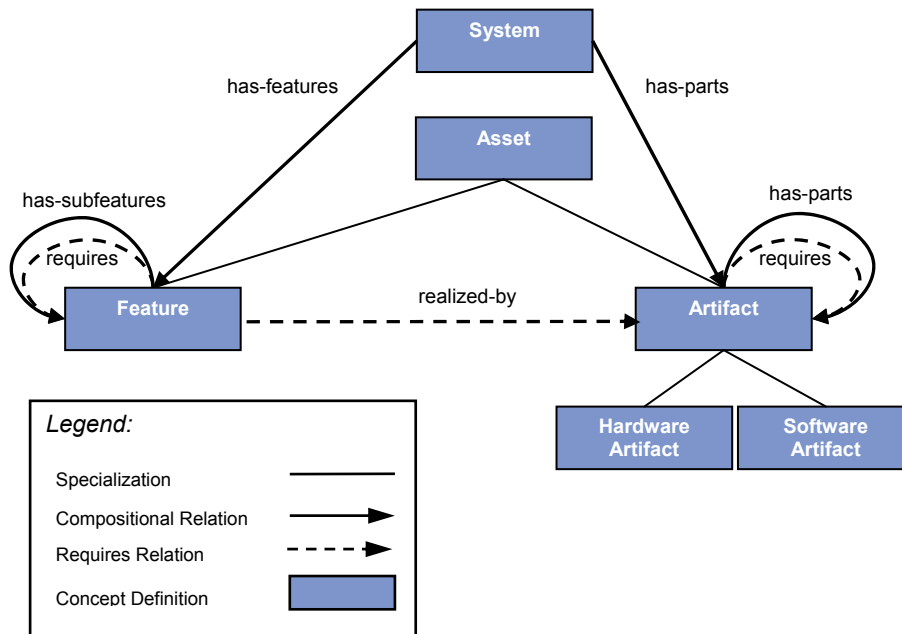
Figure 2. Common Applicable Model (CAM)

Further relations between assets that do not describe a hierarchical structure like taxonomic and compositional relations do are called *dependency relation*. A *uni- or bidirectional* dependency can be defined between concepts in arbitrary places of the ontology. Examples for this relation type are the *requires* relation for unidirectional und the *excludes* relation for bidirectional dependencies. For bidirectional relations this means that during the configuration process this relation is processed in both ways, no matter which concept participating in the relation is instantiated first. Unidirectional relations are different; they are only processed in one direction. E.g. when a software module requires the existence of a specific driver the existence or absence of this driver has no impact on that software module.

The *realizes* relation is a further bidirectional dependency that expresses that features are realized by artifacts in an *n*-to-*m* mapping: one feature can be realized by one or more artifacts and the other way round one or more features can be realized by one artifact. Several situations during product derivation can occur when handling *n*-to-*m* mappings. The easiest possibility is when a feature is given and the needed artifacts descriptions are generated. A more complex situation is given when an artifact is given (e.g. computed by a feature) that realizes a further feature. This feature has to be generated and the artifact has to be integrated. A even more complex situation is given, when a feature is realized by several artifacts like in the first situation, but one or more of these artifacts are already related to the feature, others are already generated but not yet integrated (because they are used by other features) and further artifacts are not yet generated. For those tasks a clear separation of possible features and artifacts on the one side and features and artifacts that are used for a specific product derivation on the other side is needed. In the structure-based configuration tools like KONWERK [Günter and Hotz 1999] this is realized by distinguishing between *concepts* for describing features and artifacts in general and *concept instances* for describing features and artifacts of a specific product derivation. In situations as they are described above for some features concept instances and several artifacts are already generated and for not yet generated artifacts still concept description are present.

4. PROCESS

The configuration process in structure-based configuration is an incremental process – in each step one configuration decision is made and its effects on the partial solution configured so far are computed. Thus, in an interactive configuration process some decisions are made by the user and some are inferred by the configuration system. One aim of our approach is that most of the decisions on a lower level of abstraction (e.g. decisions about components or their parameters) are inferred by the configuration system. This is possible because of the mapping between features and artifacts defined in the configuration model (see Figure 2).

The first step in the configuration process is the selection of the product one wants to configure. Next, its features are selected. For each feature the user selects the configuration system infers the hardware and software artifacts necessary to realize the feature based on the realized-by relation in the configuration model. Thus, the product configuration is more efficient because the user only has to make those decisions that cannot be inferred based on other decisions already made. Furthermore, feedback concerning the effects of decisions can be given after each decision made by the user. For example, the user can realize that a specific feature cannot be selected anymore because of earlier decisions. This is especially important for non-expert users who are not familiar with the dependencies between decisions in the configuration process. However, even for experts this is a useful support, since they cannot overlook dependencies by mistake during the configuration process.

Based on the configuration model it is possible to compute all necessary decisions needed to configure a specific product. These decisions are collected in an agenda and can be displayed to the user. Additionally, these decisions can be grouped in several *subtasks* to structure the process. One group e.g., can encompass all decisions concerning features and further groups consist of decisions about hardware artifacts and software artifacts. This structuring of the configuration process is also defined in the configuration model, in a specific part called procedural knowledge. Since a configuration process can consist of a large number of decisions it is important to guide the user during configuration. Displaying these subtasks (in a specific order) to the user is one possibility to provide guidance.

5. EVOLUTION OF THE MODEL

Products and / or product components evolve over time. New version and variants are built, and as a fact of synchronization between the configuration model and the existing artifacts, also modeled for automated product derivation. This can have an effect on the relations between the artifacts that are combined for generating specific products as well as the relations between features and artifacts. A constantly growing artifact repository increases cognitive complexity and makes it more difficult to configure products specific for a given task description.

For most customers, however, it is not of interest which versions of software modules are used in the product. The functionality plays the key role in selecting the product that best suits the customer's requirements. Therefore, it is natural that evolution of artifacts (shown in terms of version numbers etc.) should be hidden from the customer. In most scenarios, especially in embedded systems like car electronics or mobile phones, the product that satisfies a required functionality is sufficient and the version numbers of software modules are not relevant. Such a behavior can be achieved by maintaining

the mapping between features and artifacts that realize those features during evolution of the model. When for example a new version of a software artifact is generated, the mapping(s) between this artifact and the feature(s) that this module realizes are simply modified such that they now are linked with the new version. After this, during an automated configuration process the new version will be selected. This evolution of the mapping between features and artifacts is depicted in Figure 3.
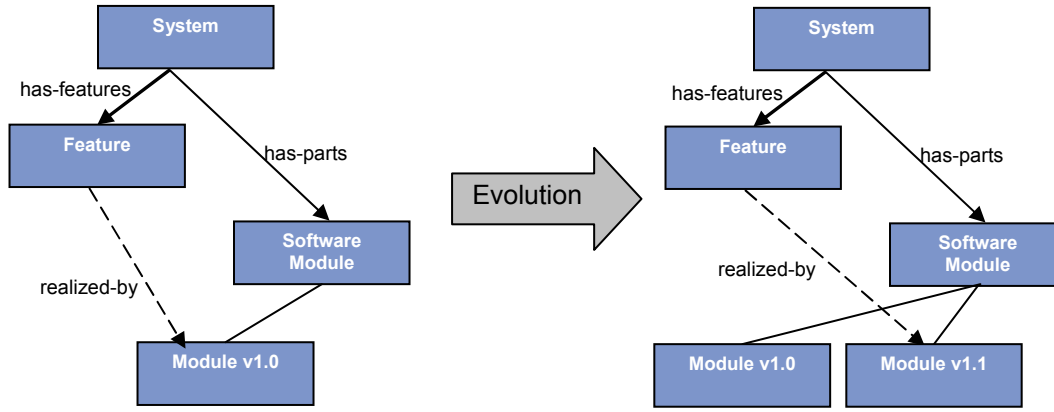
Figure 3. Synchronized Evolution of Artifacts and Mappings

Modifications to the configuration model are captured through *change operations*. *Basic* operations are those that cannot be further partitioned – i.e. simple modifications like adding a parameter value or deleting a concept definition. *Complex* change operations are composed of multiple basic operations or include some additional knowledge about the modification. They provide a mechanism for grouping basic operations into a logical unity (compare [Klein and Noy 2003]). Moving a group of concept definitions to a different superconcept for example can be split into moving each of those concepts separately. Hierarchies of change operations can be formulated to exploit the inheritance mechanism for specifying common properties.

Based on the assets and the relations that are specified in the CAM the complex change operations can be composed to force a combined modification of dependent modifications. Adding features or adding artifacts e.g. have direct impacts on the consistency of the mappings as further elaborated in the following.

- *Add feature:*
  Features describe the functionality of the product or of product components. Therefore, a feature that is not realized by some component is useless for product derivation[4]. Instead, adding a feature to the configuration model should directly entail the addition of the mapping between this feature and the artifact(s) that realize(s) it. The complex change operation can e.g. look like this:

  add feature -> (introduce feature, add mapping(s) to artifact(s))

- *Add artifact:*
  When new artifacts (in form of a concept definition) are introduced into the configuration model, they inherit the relations from their superconcept definitions. For dependencies (e.g. excludes or realizes), however, this is not always applicable. A new version of a software module may contain a bug fix and therefore no longer stay in conflict to other artifacts like the previous version did – modeled through an excludes dependency. The new version of that module usually realizes the same feature(s) the old version did, but it may also realize additional ones or be seen as a replacement for future products. For new versions (assuming that the newest version should be used) usually the mapping should be moved while for new variants (that are intended to coexist) a copy of the mapping has to be introduced. This results in the following complex change operations:

  add new artifact -> (introduce artifact, add mapping(s) to feature(s))
  add version -> (introduce artifact, move mapping(s) to feature(s))
  add variant -> (introduce artifact, copy mapping(s) to feature(s))

---

[4] We do not consider *planed features* in this paper.

The relations for the basic entities feature and artifact are predefined in the CAM and therefore can be automatically processed. Furthermore, some kind of dependency analysis is used to inspect other relations and restrictions that have been added to more specific concept definitions for building the complex change operation.

6. CONCLUSION & OUTLOOK

Using feature models has been widely implemented to improve product modeling for cognitive reasons [Felix et al. 2001] (documentation and customer-sales scenarios), product derivation [Hein et al. 2001; Kang et al. 2002] (supporting developers and enhancing reuse strategies) and configuring products in a more customer-oriented way [Ardissono et al. 2002; Felfernig et al. 2002] (e.g. in internet scenarios).

For feature-based configuration models, complex change operations can give the needed means for achieving that modifications to features and / or artifacts and mappings between features and artifacts are always processed together. Therefore, in a tool that supports evolution of configuration models, the modification of a feature or artifacts cannot be done independently of inspecting the relations this artifact participates in. Instead, complex change operations are composed that always have to be processed as one operation. Thus, the user can be forced to also modify the corresponding mappings or in a more sophisticated reasoning engine this might be (at least partially) automated.

The underlying dependency analysis is expected to be also usable for similar tasks like innovative configuration. This method can be used in conflict situations – i.e. in situations where the configuration solution generated so far, the configuration model and the task specification are not consistent. *Innovative configuration* is the task of extending the configuration model dynamically in the configuration process such that newly introduced concepts or concept properties enable additional solutions to the given task [Hotz and Vietze 1995]. A danger here is that dependencies to other concept definitions and configuration decisions can be violated.

7. ACKNOWLEDGMENTS

8. REFERENCES

1. Ardissono, L., Felfernig, A., Friedrich, G., Goy, A., Jannach, D., Meyer, M., Schäfer, R., Schütz, W. and Zanker, M. (2002). Customizing the Interaction with the User in Online Configuration Systems. Proceedings of the Configuration Workshop on 15th European Conference on Artificial Intelligence (ECAI 2002), Lyon, France, pp. 119-124.

2. Brüne, S., Halmans, G. and Puhl, K. (2003). Modelling Dependencies between Variation Points in Use Case Diagrams. Proceedings of the 9th International Workshop on Requirements Engineering Foundation for Software Quality (REFSQ), Klagenfurt, Austria.

3. Felfernig, A., Friedrich, G., Jannach, D. and Zanker, M. (2002). Semantic Configuration Web Services in the CAWICOMS Project. Proceedings of the Configuration Workshop (ECAI 2002 Workshop), pp. 82-88, Lyon, France.

4. Felix, D., C. Niederberger, P. Steiger, and M. Stolze (2001). Feature-oriented vs. Needs-oriented Product Access for Non-Expert Online Shoppers. Proceedings of the 1st IFIP Conference on e-commerce, e-business, and e-government (I3E), pp 399-406.

5. Ferber, A., Haag, J. and Savolainen, J. (2002). Feature Interaction and Dependencies: Modeling Features for Re-engineering a Legacy Product Line. Proceedings of 2nd Software Product Line Conference (SPLC-2), San Diego, CA, USA, pp. 235-256.

6. Günter, A. (1995). Wissensbasiertes Konfigurieren. Infix, St. Augustin.

7. Günter, A. and Hotz, L. (1999). KONWERK – A Domain Independent Configuration Tool. Proceedings of Configuration (AAAI 1999 Workshop), Orlando, Florida, USA, pp. 10-19.

8. Hein, A., Schlick, M. and Vinga-Martins, R. (2000). Applying Feature Models in Industrial Settings. In Software product lines - Experience and research directions (Ed Donohoe P.). Kluwer Academic Publishers, pp. 47-70.

9. Hein, A., MacGregor, J. and Thiel S. (2001). Configuring Product Line Features. <u>Proceedings of Feature Interaction in Composed Systems (ECOOP 2001 Workshop)</u>, Budapest, Hungary.

10. Hotz, L. and Vietze, T. (1995). Innovatives Konfigurieren in technischen Domänen. <u>Proceedings of 9. Workshop Planen und Konfigurieren (PuK 1995)</u>. DFKI Saarbrücken, Kaiserslautern, Germany.

11. Hotz, L. and Krebs, T. (2003). Supporting the Product Derivation Process with a Knowledge-based Approach. Proceedings <u>of Software Variability Management (ICSE 2003 Workshop)</u>, Portland, Oregon, USA.

12. Kang, K., Cohen, S., Hess, J., Novak, W. and Peterson, S. (1990). <u>Feature-oriented Domain Analysis (FODA) – A Feasibility Study</u>. Technical Report CMU/SEI-90-TR-021, Carnegie Mellon University, Pittsburgh, PA, USA.

13. Kang, K. and Lee, J. and Donhoe, P. (2002). Feature-oriented Product Line Engineering. <u>IEEE Software, Vol. 7 (8)</u>, pp. 58-65.

14. Klein, M. and Noy, N. (2003). <u>A Component-based Framework for Ontology Evolution.</u> Technical Report IR-504, Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands.

15. Krebs, T. and Hotz, L. (2003). Needed Expressiveness for Representing Features and Customer Requirements. <u>Proceedings of Modeling Variability for Object-Oriented Product Lines (ECOOP 2003 Workshop)</u>, Darmstadt, Germany.

16. Robak, S. and Franczyk, B. (2001). Feature Interaction and Composition Problems in Software Product Lines. <u>Proceedings of Feature Interaction in Composed Systems (ECOOP 2001 Workshop)</u>, Budapest, Hungary.

KEY PRINCIPLES FOR SALES CONFIGURATION DESIGN

Fabrizio Salvador[1] and Cipriano Forza[2]

[1] Department of Operations and Technology Management
Instituto de Empresa
Maria de Molina, 13 B
Madrid, Spain 28006
Email: { Fabrizio Salvador, fabrizio.salvador@ie.edu }

[2] Department of Mechanical and Civil Engineering
University of Modena and Reggio Emilia
Via Vignolese, 9005/b
Modena, Italy 41100
Email: {author_name, forza.cipriano@unimo.it }

By offering differentiated products a firm may be able to increase product price or market share because of the better fit of its products with what customers demand. However, the decision to proliferate the product offer, or to just allow customers to define more and more product features does not automatically lead to greater customer satisfaction. In fact a wide assortment of product variants and options may end up confusing the customer. Ultimately, while trying to please its customers by offering increased product variety, a firm may end up confusing or annoying its customers, thus loosing potential sales. In other words, the firm faces a "product variety paradox".

Recent developments in product configurators appear to offer new solutions to the front office supporting the interaction with the customer when custom products are offered. To date, however, we still know very little of the potential of such class of software products to reduce the "product variety paradox".

The present paper analyzes the underlying principles on which successful sales configurators have been built. In doing so, the paper attempts a formalization of the mechanisms through which a firm s product assortment can be efficiently and effectively presented to the customer thus increasing a firm s commercial success.

1. INTRODUCTION

The assumption underlying variety-intensive product strategies (Sanderson and Uzmeri, 1995) is that customers have idiosyncratic needs. Consequently, by offering differentiated products a firm may be able to increase product price or market share because of the better fit of its products with what customers demand. However, the decision to proliferate the product offer, or to just allow customers to define more and more product features does not automatically lead to greater customer satisfaction and, hence, to greater commercial success. In fact, a wide assortment of product variants and options may end up confusing the customer, as he/she would experience high cognitive complexity in evaluating product alternatives (see Huffman and Kahn, 1998). In other words the firm may experience the "productivity paradox": offering more variety to increase sales may lend to loss of sales. From a theoretical point of view this phaenomenon may be framed in cognitive terms.

2. SETTING THE PROBLEM: COGNITIVE COMPLEXITY AND PRODUCT VARIETY

In order to better satisfy the customer through differentiated or personalized products, the customer must effectively individualize, among the different variants offered by the firm, the option that satisfies his requirements. Let us consider, for example, the difficulties we have when selecting a piece of clothing in a big department store: the proliferation of different models, colors, trademarks, etc. can exert a negative influence and become an obstacle for our decision to purchase. This kind of problem is not exclusive of the clothing sector. On the contrary, it may appear every time we have to select the product variant that suits our preferences among "n" possible variants. In this case, the buyer faces the so-called

cognitive complexity; he finds difficulties in understanding what is offered and, in particular, how the products differ as far as utility. Even in the presence of only three alternatives (A, B and C) it is difficult to decide, considering their utility, which one is the best. What characteristics of A satisfy my needs better than those of B or C? If in some aspects A seems better than B, but in some others the contrary is true, which one should I choose? This kind of cognitive obstacles do not help the firm that adopts a strategy of variety, since such variety may confuse the customer rather than convince him that the firm has a solution for his problem.

Even if we admit that our potential client does not have great difficulties to determine whether the variant A is better than the variant B or C, the problem of cognitive complexity is not completely solved yet. Once the customer decided that, for example, the variant B is better than A and C, there is still a doubt: "Isn't there, among all the other variants offered, one that could be better than B?" When the variants are extremely numerous, to compare them as we did with A, B and C becomes an impossible and costly task. The risk is that the customer may delay his decision to purchase, not because he finds it complicated to choose among the variants, but because he does not have a complete knowledge of all the possible alternatives.

In synthesis, this is a paradoxical situation. On the one hand, the firm increases the product variety in order to capture a bigger number of customers or to obtain higher prices. On the other hand, the decision to offer a great variety may mislead the client or make the process of selection so difficult that finally the client renounces. The problem of cognitive complexity is well known by the firms that offer variety and personalization. Generally, the solution is based on the sales force that can make the acquisition process easier for the client.

Coming back to the example of the clothing department store, the role of the shop assistant is to help the customer find, among the chaos of garments offered, the variant that best suits his requirements. But, why is the shop assistant useful to individualize what we are looking for? To answer this apparently simple question, we must compare our behavior in two different situations: 1- when we are alone, looking for the product and 2- when the shop assistant helps us to look for the product. In the first case, we examine, sequentially and systematically, the garments in the way they are displayed (obviously in the corresponding area). At a glance, we quickly verify if each piece of clothing can suit our needs. In the second case, instead, we tell the shop assistant what we are looking for, adding a series of attributes: a shirt with vivid colors, made of cotton, button-down, backstitched, etc. The shop assistant considers the options the shop can offer, and then proposes us a number of shirts. In other words, he minimizes the complexity we are supposed to face if we want to understand what the shop is offering. We already know that the type of shirt that interests us - if the assistant has not forgotten any option - is in front of us, thanks to the previous selection made by the salesman. The complexity of the selection process was eliminated because the variety of product was presented by attributes (color, stitch, fabrics, etc) rather than by alternatives (shirt # 1, # 2,..#87 from a shelf 10 meters long) and because we trusted the shop assistant for a first skimming of the products offered by the department store.

To understand how the representation of a set of product variants influences the cognitive complexity in the selection process of a potential customer, two marketing researchers chose 78 students for an experiment. They had to select a sofa for the house where they would live after graduating. In the first part of the test, the students had to understand the product offer. The results was that the representation of the offer in terms of attributes, rather than in terms of alternatives, helps the customer's understanding of the offer and enables him to express his specific requirements and choose a satisfactory variant.

This experiment demonstrates that the description of product varieties in terms of attributes, rather than in terms of alternatives: 1- allows the firm to transmit, with considerable efficiency, the offer to its customers and 2- enables the customer to express, in a more precise way, his own requirements.

3. AIMS AND METHOD

Recent developments in Information and Communication Technology made available a class of software products, often termed as "product configurators", which appears to offer new solutions not only to the back office automatically generating technical product documentation but to the front office as well supporting the interaction with the customer when custom products are offered. To date, however on, research product configuration has been focusing especially on the back-office advantages of such solutions, while we still know very little of the potential of such class of software products to reduce the "product variety paradox".

As mentioned before, the sales personnel typically carry out this activity of "education" of the customer or of "support to selection". The learning process, however, may be long and repetitive, involving high commercial costs. There is still a problem, how to enable the customer to "teach himself" or auto-configure the product", at least partially. To fulfill this aim, it is necessary to design a supporting program that describes the product, from the commercial point of view. The planning of this support requires a series of fundamental actions and choices which will be analysed throughout the paper:

1. choose methods for product description

2. delimit the space of the possible choices on the part of the customer

3. communicate how different options may create values for the customer

4. structuralize the ways in which the customer learns and/or defines the product characteristics

5. foresee how the interaction between the customer and the commercial support facilitates the learning process and, therefore, minimizes the cognitive complexity faced by the customer.

From a methodological standpoint, the following sections of the paper analyze the underlying principles on which successful sales configurators have been built, based on anecdotal evidence present in the literature or on the analysis of sales configurators present on the web. In doing so, the paper attempts a formalization of the mechanisms through which a firm s product assortment can be efficiently and effectively presented to the customer. We then discuss how these mechanisms, as a whole interact and how they can increase a firm s commercial success. Finally, we speculate about the possibility of extending the proposed mechanisms outside the scope of designing sales configurators, proposing that they can be taken as general principles to describe efficiently and effectively a firm s product assortment.

4. DESCRIBING THE PRODUCT

There is not only one way of representing a generic product to the customer. The product complexity, its importance for the customer and the customer's availability (or not) to get information about the product are factors that contribute to determine the most suitable method to describe the product, as far as its commercial configuration. To understand how different descriptions of the same product differ one form the other, we may consider the simple case of automobiles (See Figure 1).

Figure 1. Product descriptions with different degrees of abstraction
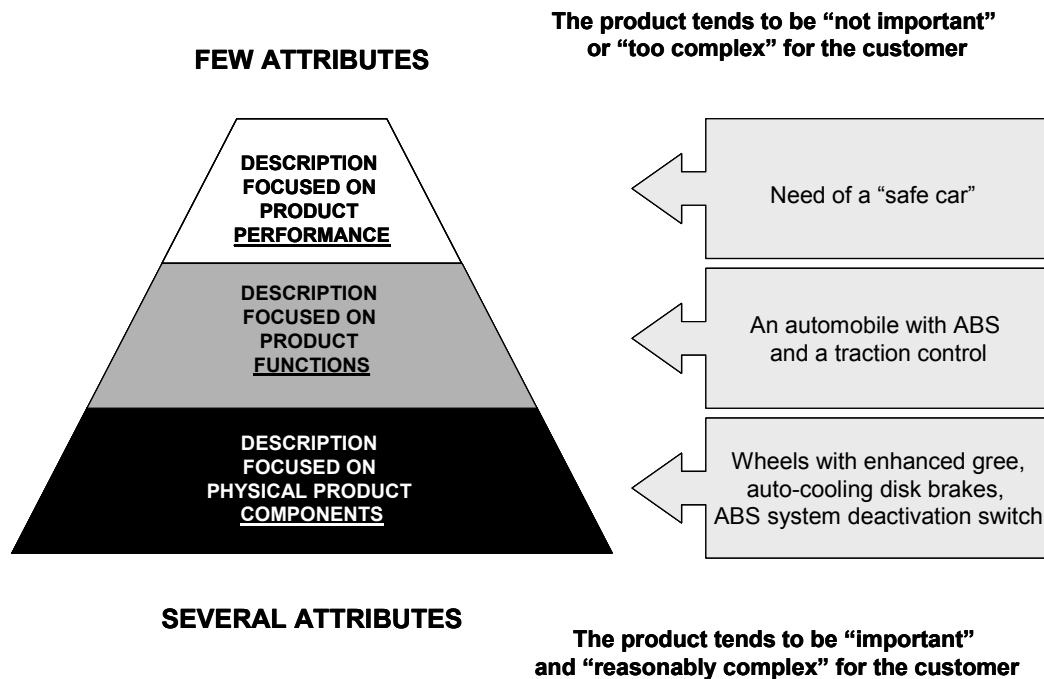
Some of the possible buyers of automobile are interested in some fundamental performance, without caring about which functions or systems are required. For example, a grandmother who wants to change her car or the head of a family who is not interested in reading specialized magazines, may only ask for an "economical and safe car". A more expert buyer may probably require an automobile with ABS or traction control. In other words, the customer speaks in terms of functionality, rather than in terms of performance: the wheels should not be blocked when braking or accelerating. A more expert buyer, one who knows how the car is made, may ask for an automobile assembled with a special kind of wheels, to enhance grip,

or he may prefer a model with auto-cooling disk brakes rather than with the traditional single disk brakes. In the case of a really expert driver, he may even ask the concessionaire to install a switch to deactivate the ABS system when very skilful driving is needed, for example under bad road conditions, due to snow or frost. The third type of customer, in synthesis, describes the product in terms of components rather than in terms of functions.

The example of the automobile shows us how the same product can be described in very different ways. Different product descriptions can be lined up on a one-dimension continuum. On one end of the line, we can place synthetic descriptions focused on performances and on the other end detailed descriptions focused on product components. In the intermediate position we find the descriptions focused on the functions.

It is important to notice that the three situations are compatible with product descriptions based on attributes. The change is in the nature of the attribute: performance, function or components.

An example of different product descriptions is given by the notebook selector of ZDnet and by the Chl personal computer configurator. In the first case, it is not a configurator, but a selector that starting from a series of characteristics, singles out one or more suitable product variants. Anyway, the elaboration of a commercial model or of a formal description of the product, from the commercial point of view, are activities performed by the selector or by the product configurator. In the case of ZDnet (www.ZDnet.com) the firm has conceived a product description at a very abstract level, to help the inexpert potential customer to choose among the numerous variants offered. The selector asks general questions such as: how much are you willing to spend? How often do you take a plane, situation where weight is an important factor? If weight is not a problem, are you looking for a notebook that can be used as a substitute for a desktop PC? These simple questions, that anyone can answer, allow the customer to skim the offer, reducing the number of options to a dozen of potential products. On the other side of the ZDnet selector, we find the personal computer configurator created by Chl (www.chl.com). In this case the product description is made at the level of single physical components, that must be entirely specified, from the computer case to the motherboard to the possible peripheral units. It is necessary to have considerable knowledge of the components and functions of a personal computer, to get an effective advantage from the possibility of expressing one's preferences at a product component level.

To choose the appropriate way to describe a product is not a question of using a language that could be more or less abstract. In some cases, the product characteristics require an input for the commercial configuration that has a format different from the textual one, up to now considered. For example, a firm that sells cooling systems for cold stores, has as a fundamental input for the commercial configuration process, the layout of the cold store: the plan of the room, free walls on which the cooling fins can be placed, possible obstacles for the installation. A description in terms of measures, using a standard format to collect the necessary information, would be practically impossible! In general, the problem of getting the characteristics in the form of a layout is common to all the firms that manufacture systems. The alternatives for these firms are two: 1- to limit the configuration only to system components; leaving to sales engineering the layout definition, 2- to invest in the development of an instrument for information acquisition through layouts. In this case, the company has to compare very carefully, the benefits of automation with the costs for the development of a software solution that, with great probability, will require a certain degree of personalization, and therefore a further investment.

Another important aspect of commercial configuration is the total description of the configured variant. Obviously, this requirement is based on the fact that the final aim of a commercial configuration is to carry out an economic transaction. Price and an indicative delivery time are fundamental outputs of the commercial configuration process. To define all the aspects of the product is very important to avoid misunderstandings, opportunistic behaviors, disputes, etc. Furthermore, it is essential to give a complete vision of the product ordered by the potential customer, because in this way he is able to verify at a glance, the whole effect of his preferences, considering even some interactions that may be neglected due to his choice by attributes. Let us consider, for example, the case of the sofa, described in the first paragraphs. If we offer the client the possibility to choose the type of wood needed for some parts of the sofa and the color of the fabric, surely, he will be interested in knowing whether both colors, put together, produce a pleasant match or not. This problem could have been avoided if the choice had been made by alternatives, showing each one with its corresponding illustration. The sites of some automobile factories, that allow the customer to simulate the matching between the colors of the interior and the body, avoiding unpleasant delusions when the automobile is taken from the car-shop, offer this kind of service.

5. DELIMITING THE OPTIONS

Even if the most suitable way to describe the product has been determined, the problems associated with the definition of the commercial model have not finished yet. A typical dilemma is to decide whether to include in the commercial model all the possible variants. From a productive point of view, it is important to remember that, even if the various components needed to create an offer of "n" products have been already designed, to include all of them in the commercial model means to assure the supply to a possible client, of a variant that is very rarely manufactured. This may bring about a series of difficulties in the supplying, planning and control of the productive activities, with the risk that the costs may be higher than

the profit obtained by selling that kind of variant. In synthesis, it is necessary to remember that behind a commercial model there is always a workshop that has to manufacture the product!

Furthermore, the reasons for limiting the variants among which the customer can choose, in relation to the ones the firm can offer, may be purely of commercial nature. In fact, to offer many variants, at the end, always complicates the commercial model as well as the customer's choices. Sometimes it can be more practical to label some "exotic" options as not available, in order to reduce the quantity of information the customer has to supply to obtain a complete configuration. The simplification of the configuration dialogue - a further confirmation of the problems associated with the cognitive complexity of the configuration tasks - is one of the requirements most frequently underlined by the commercial department of the firms that are implementing a product configurator. The customer who asked for something that is not foreseen by the product model should get in touch with a salesman and check the possibility to obtain the requested variant.

In other cases, the decision to delimit the variants among which the customer can choose, is a consequence of the rationalization of the offer the firm makes, when it decides to structure the configuration process. Very frequently, the company "realizes" that different product families, in some way, tend to overlap. The most expensive variants of the family at "entry level" or with "low power" overlap with the cheapest variants of the immediately superior family. This may happen even in the most famous companies. Boeing, for example, expanding the capacity of the 737 family and developing more spacious and comfortable versions of the 767 family, finished up by covering, with these two families, the field of the 757 family (that never actually took off"!) In general, if the overlapping of product families is not commercially justified, the formalization of the dialogue should reasonably lead towards a limitation of the variety offered, in relation to the variety the firm is potentially able to offer (and in the future towards a new definition of the latter).

A third possibility to reduce the quantity of commercial characteristics to be specified in the product configuration is the pre-determination of some characteristics that generally are requested in a certain standard. In other words, to foresee default values for some product characteristics that rarely acquire values different from those pre-established. An extreme case of this approach is to offer the customer a set of pre-configured products, with the possibility to modify the attributes according to his preferences, as long as he respects the links defined in the commercial model.

Figure 2 summarizes the principles that can be applied to limit the options offered through a commercial dialogue thus leading to a simplification of the commercial model and to a reduction of the cognitive complexity.

Figure 2. Simplification of the commercial model by limiting the options
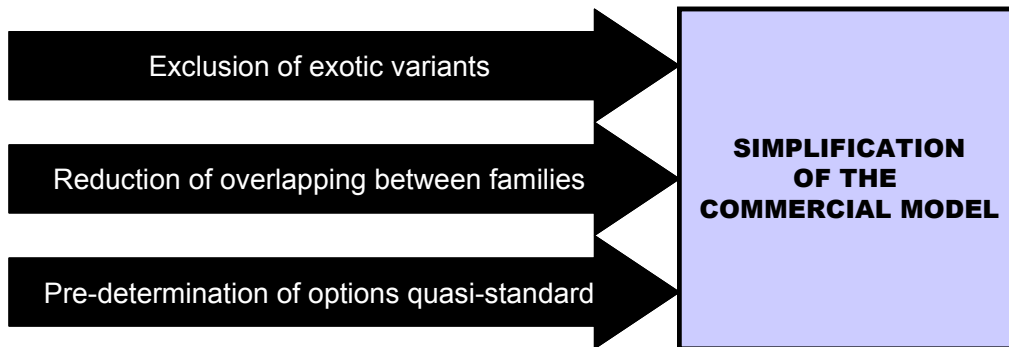
6. COMMUNICATING THE VALUE

As mentioned before, any kind of strategy of product proliferation develops different options that create different functions, as a more precise answer to the preferences expressed by different customers. To communicate the value of such options, we can certainly describe the product attributes using a language that the customer can easily understand, we can avoid overwhelming him with too many options, but this is not enough. Maybe the customer simply wants to understand the product or the way in which the different attributes determine the functionality of the product. This is a typical activity performed by the commercial staff. It is a complex function, because it somehow implies a certain "didactic" skill on the part of the salesman, and because the learning process on the part of the customer may be long. Moreover, and above all in the case of complex products and/or subject to rapid technological evolution, it is necessary to teach the different product functions not only to the customer, but also to the salesman. To develop automated solutions to illustrate the functions associated with the different product attributes, on the one hand, enables the client, as well as the salesman, to learn autonomously, on the other hand, the learning process is carried out according to the modes and times of the user, maybe when he is relaxed at home or during a calm moment at work. Better knowledge will give the customer better elements to decide about the product he is willing to buy.

If the product presented by the firm is neither valid nor suitable for the customer, the firm that implements this approach will lose a potential client. But, if the product is not suitable and the customer decides to buy it, at the end the firm will also lose the client.

The progress of multimedia and the growing bandwidth of telecommunication services offer a number of promising opportunities to the firms that want to communicate the values of their product varieties. Films, animations, graphics and sounds help to reduce the time the customer needs to understand the complexity of a product family and consequently increase the profits of strategies of product proliferation.

To communicate the value of the variety offered to the customer does not mean only to explain what the different alternatives are able to do (See figure 3). All the alternatives are not equally easy to elaborate. Some of them may request the design of some ad-hoc components, and are part of a field that is out of the product configuration process. However, due to some commercial reasons, the firms cannot exclude from their product offer those semi-configured or particularly problematic variants. In these cases, to communicate the value of his specific choice to the client, also means to make him aware that he is asking for something "special", and that his requirement most probably will influence the price and/or delivery times.

Figure 3. Fundamental activities to communicate the value of the variety offered by the firm

$$\text{VALUE} = \frac{\text{NEED SATISFACTION}}{\text{COST \& TIME}}$$

Communicate how each option contributes to satisfy a specific need or functionality

Communicate implications on price and delivery time of each choice

Dell Computers, one of the first companies in the world to sell configurable products online and one of the biggest producers of PC and portables, provides an example of "customer's education" aimed at optimizing the utility of the available options that is perceived by the client. If we take, for example, the option of hard drives in the main configuration dialogue, we see that there are a number of variants. If the customer is not ready to decide yet, he can ask for more information (learn more). At this point he is able to consult three lists. Two of them have an exclusive educative function: one explains the primary characteristics of hard drives and how they determine functionality, the other describes one by one all the technical attributes of the hard drive. The third list combines the information obtained with what the firm offers, supporting the selection process of the potential client. Finally, but not of minor importance, another benefit of customer s education is that the potential client, after getting an idea of what product could satisfy his needs, can go back to the main menu and evaluate if price variations are justified.

7. STRUCTURING THE PROCESS OF INTERACTIONS WITH CUSTOMERS

The elaboration of a commercial model and the definition of how the commercial dialogue will be carried out raises - apart from the problems mentioned in the previous paragraphs – the issue of how to structure such interaction. In the simplest case, the answer to this question implies the definition of the order according to which the different questions are asked. A firm that wants to communicate its customers the idea that it offers strong customization, for example, may consider useful to present, in the first place, the product attributes that offer more possibilities of choice. For a firm that sells tailored shirts, the type of pattern (checked, striped, plain, etc.) and the different variants (big or small squares, tartan, etc.) will surely be the first questions to ask. In other cases, especially for technical products, the sequence of questions has to follow, as close as possible, the order the customer typically uses when describing or specifying the product. For example, a firm that offers personalized pumps, firstly, will ask the type of application (submergible, peripheral, etc.) And then, some fundamental data such as flow rate, discharge head and so on. In general, the idea is to allow the client to search the variety offered by the firm in the most natural and spontaneous possible way. The importance of this condition is not reduced if the user is the salesman. In fact, if the salesman perceives the process followed to define the product characteristics as something complicated and unnatural, the possibilities to implement a successful solution, even if partially automated, are very low. The elaboration of a structured process that leads the customer, probably with the help of the salesman, towards the

definition of a commercial configuration, presents some difficulties that derived, not from the customer, but from links between different product attributes. For example, in the case of a utility vehicle, the option air conditioning" may not be available for the version with a reduced cubic capacity, due to the fact that excessive power will be absorbed by the cooling compressor from the crankshaft. The presence of links between options determines a sort of rigid order in the commercial dialogue. For example, some questions must be asked following a fixed sequence. Let us take the case of a scooter: if we want to configure a scooter with the options "Country" and "double-seat", we must ask first the option "Country" and then the option "double-seat", because in some countries two people are not allowed to ride on the same motorcycle while in others it is permitted. A second rigid condition, related to the first one, is the fact that possible choices of a certain attribute depend on previous choices of other attributes. For example, if in the configuration of a bicycle, we have selected the options " titanium alloy frame" and "double damper", it is evident that the choice of different types of rims will be limited to those that respect or surpass a certain minimum value of strength. In some cases, these limitations finally eliminate any kind of freedom while choosing certain attributes: only one of the levels admitted for a certain attribute could be compatible with the choices previously made in the commercial dialogue. Anyway, the presence of links between product attributes on which the customer can express his preferences, generally does not exclude the possibility to define various alternatives. Let us consider again the example of the bicycle. For the attributes "type of frame" and "weight" the customer could specify the maximum weight, and so a series of "heavy" component variants would be automatically excluded from the choice (frames of chromo-molybdenum steel, standard saddles and other elements) On the contrary, the weight could be simply calculated by adding all the weights of the components selected. In this case, in order to reach the required weight, it may be necessary to modify the choices made. It would be interesting to allow the customer to start the commercial dialogue from any compatible attribute of the product. Yet, this complex option is rarely found in commercial applications. In fact, the most common approach consists in considering a precise sequence of questions and answers, in which, as shown in Figure 4, we can have:

1. questions that depend on the previous answers

2. possible answers that depend on the previous answers

3. algorithms that develop answers on the basis of previous preferences

Figure 4. Structure of the commercial dialogue
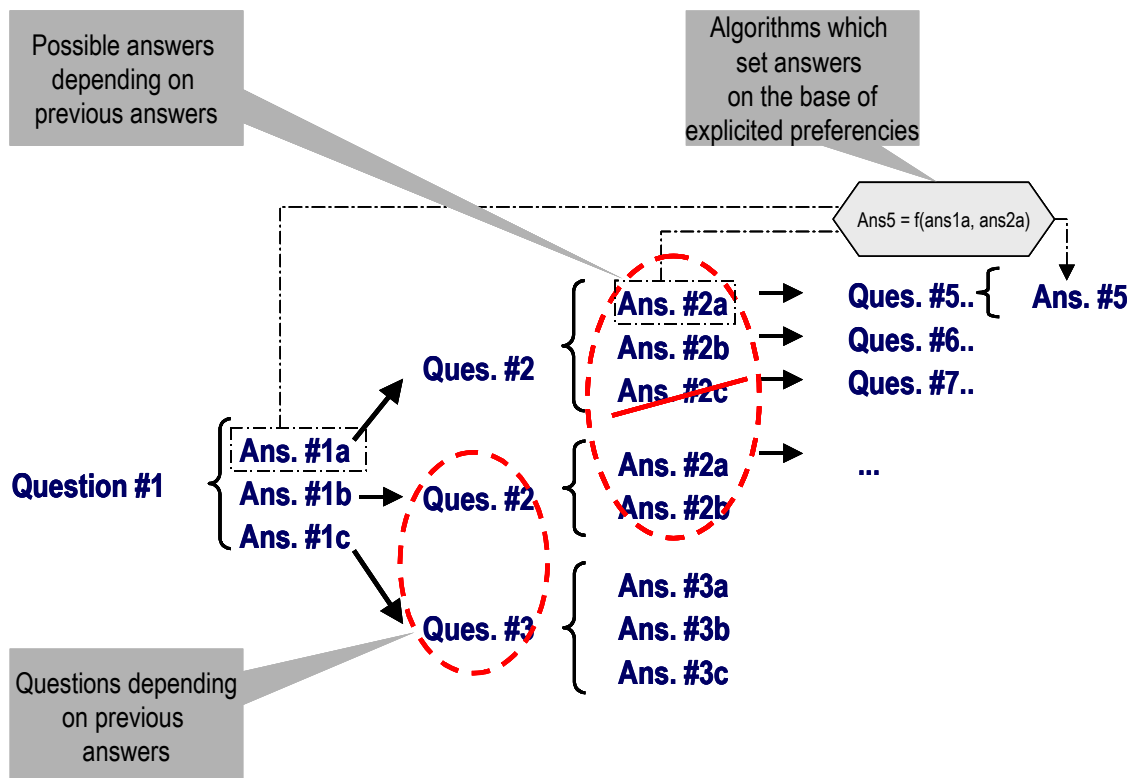
To represent the commercial dialogue as a decision tree is a conceptual approach to the planning of the commercial model. The way, in which this representation will be implemented in the product configurator, obviously depends on the particular software system adopted.

Methods for product description, delimitation of customer's choices, communication of the value related to the different options and finally, structure of the commercial dialogue, sum up the actions needed to build a good commercial product. We still have to consider the fact that a firm has to serve, with the same product family, many types of different customers (let us take the example of automobile factories) The question is: Is it possible and efficient to serve all these types of clients using a single commercial model and, consequently, using a single commercial dialogue? or is it better to develop different commercial models according to different types of customers? There is not a correct answer. First, it is fundamental to consider marketing aspects and the investment necessary to develop different commercial models and then, to determine if and to what extent, one solution is more appropriate than the other.

8. INTERACTION AND LEARNING

The reduction of the complexity of the product offer, that the customer notices, as mentioned before, is reached not only by diminishing the information load for the client, but also by increasing his ability to understand and evaluate the available product information. In other words, learning is a fundamental activity that reduces cognitive complexity. Learning is a process, i.e. it consists of a series of actions carried out during a period of time. The salesman, who visits the customers now and then, gives him product documentation and communicates with him in different ways, feeds a learning process on the part of the customer. A solution for product configuration, that can be - at least partially- automated, will be successful if it is able to support the customer's learning process. Fortunately, interaction is in some way, an intrinsic characteristic of computer applications. That is the reason why the customer can use a commercial model as a base for his progressive learning or updating, related to product information. In order to understand the product, the structure of configuration dialogue can be consulted several times, giving different answers to see how the final product configuration changes. A better understanding of the product, eventually, enables the customer to appreciate how different possible attributes contribute to create a configured product variant that can satisfy his requirements.

The configuration of a truck offers an interesting opportunity to understand the fundamental role played by interaction, in the comprehension of how the choices we make are interdependent (product knowledge) and how these choices influence the general performance of the product. The truck studied presents a considerable number of possibilities: cabin type (7), engine (4), class (4), chassis type (2), wheel configuration (11), chassis height (4), suspension (3), power (11). Furthermore, the client can express his preferences in terms of maximum price, level of comfort, versatility, performance and fuel economy. (see figure 5) All these items are interdependent.

Let us start from the basic configuration provided by the system, characterized by a "normal" level of comfort. Let us suppose that we want to purchase a more comfortable vehicle, so we specify a "very high" level of comfort. Immediately the system offers us (associated to configuration B) a modified outlook of the performance, different from the one shown in association with configuration A. In the new table (provided with configuration B) we can see that the increase in the level of comfort implies a decrease in the level of performance (Low) while the level of versatility does not change (Normal) and the level of fuel economy is the same (High). The price of configuration B is 15% higher as well.

At this point, it is interesting to see what happened at a level of components, that is to say which components guarantee high performance in terms of comfort. The system shows that the option of very high comfort has modified the cabin type (CP 19 instead of CT 14), the chassis height (low instead of normal) and the suspension (air instead of leaf suspension).

Exploring these interdependences, the potential customer can get a clear idea of what the firm offers and above all, of what the firm can do for him. We must remember that a family of heavy trucks may have thousands of possible configurations, considering the detailed specifications.

9. CONCLUSIONS

The above reported analysis of successful sales configurators allow us to identify some key principle to guide firm in developing sales product configurators capable to present efficiently and effectively the firm's product assortment. Figure 5 summarizes these principles.

The starting point is, as repeatedly mentioned before, to reduce the cognitive complexity related to the efforts the potential customer has to make, in order to understand what the firm offers and how the different solutions can satisfy his requirements. The mainlines to design the interaction with the customer are: 1- describing the product with the abstraction level appropriate for the target customer, 2- delimiting the options the customer can make, 3- communicating the value of the different alternatives, 4- structuring the customer's process of interaction. The possibility, for the client, to interact

directly with the commercial model or through the salesman, generates a learning process that facilitates the understanding of the offer and relates it to the customer's specific needs.

Figure 5: Integrated scheme for designing customer interaction when selling configurable products
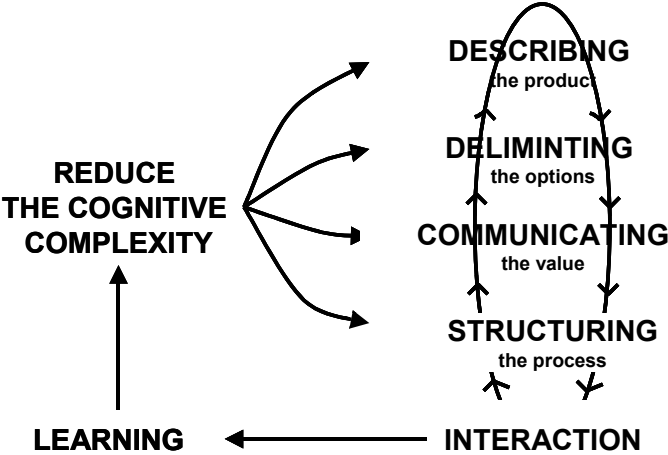
10. REFERENCES

Bibliography available form authors upon request.

XML-BASED DATA EXCHANGE OF PRODUCT MODEL DATA IN E-PROCUREMENT AND E-SALES: THE CASE OF BMECAT 2.0

Volker Schmitz, Joerg Leukel

Oliver Kelkar

Department of Procurement, Logistics and Information Management
University of Duisburg-Essen (Campus Essen)
Universitaetsstrasse 9
45359 Essen, Germany
Email: {volker.schmitz, joerg.leukel }@uni-essen.de

Competence Center Electronic Business
Fraunhofer Institute for Industrial Engineering
Nobelstrasse 12
70569 Stuttgart, Germany
Email: oliver.kelkar@iao.fhg.de

While the exchange of product model data is standardized by the STEP standard, web-based e-procurement and e-sales show a different picture. The rise of XML has led to a variety of different, often from-the-scratch catalog standards. Heterogeneity is also reflected in diverse models for product data. This is especially true for complex product models as necessary for configuration. However, the capabilities of XML-based standards are limited regarding configuration requirements. Concerning this situation, we will present results of the BMEcat standardization. BMEcat is the leading standard for catalog data exchange in Europe. Whereas its present version was the result of a company-focused standardization process, the enhancements of BMEcat 2.0 are also based on comparative analysis and research activities. We will describe the process of adopting and integrating existing concepts from literature, and the introduction of new concepts regarding price definition based on formulas and repositories for industry-wide price parameters.

1. INTRODUCTION

While the exchange of product model data is, in general, standardized by the STEP standard, web-based e-procurement and e-sales show a different picture. The rise of XML as the common format for exchanging electronic product catalogs has led to a variety of different, often from-the-scratch exchange standards. For instance, cXML and xCBL are driven by major software companies, EAN.UCC and OAGIS are being developed by industry consortia, thus non.-formal standardization bodies. Numerous vertical initiatives like PIDX complement these standards. The heterogeneity caused by a number of different exchange formats also reflect in diverse capabilities for representing product data. This is especially true for complex product models as necessary for product configuration systems. Recent studies have shown that the capabilities of XML-based catalog standards are limited regarding configuration requirements [1]. Therefore, the current state of relevant standards is not sufficient to support the increasing requirements of buy-side procurement, electronic marketplaces, and supplier participation.

Taking the described situation in mind, this paper will present results of the BMEcat standardization initiative to overcome deficits in XML-based catalog data exchange. The non-proprietary BMEcat format is the leading standard for catalog data exchange in Europe [2]. While its present version 1.2 was the result of a company-focused standardization process, the enhancements of BMEcat 2.0 are also based on comparative analysis and research activities. In the respective field of configuration, we will describe the process of adopting and integrating existing concepts from literature (e.g., bills-of-material, constraints). In addition, we will introduce new concepts regarding price calculation based on formulas and repositories for industry-wide price parameters.

Widening the scope of web-based procurement and sales systems beyond MRO goods (maintenance, repair and operations) will help to step into the next phase of business-to business e-procurement. This will enable covering complex goods and targeting new product domains and branches of industry as well, thus the business value of e-procurement systems will be enhanced.2nd and all subsequent Paragraph indented as here with no line spacing before the line.

2. PAPER ORGANIZATION AND RELATED WORK

This paper aims at introducing basic concepts for product configuration and preliminary XML data models as they will be implemented in BMEcat 2.0. It supplements previous work on the capabilities of relevant XML-based catalog standards by providing a practical solution to important problems as they are seen from the view of web-based e-procurement and e-sales systems. Our paper is structured as follows:

First we will describe three complexity levels; these are fix, parameterizable and configurable products (Section 3). Then, we will look at current practice in handling product data in e-procurement and e-sales. On one hand, we have to consider sell-side systems that are loosely coupled with buy-side systems (Section 4.1). In this scenario, the exchange of product data is limited to the minimum; hence configuration processes take place on the sell-side only. On the other hand, if companies exchange product data for e-catalog applications, we have to consider relevant XML-based exchange formats. Hence we will refer to a comparative analysis that we have done within a European standardization project [3] (Section 4.2). Both, the levels of complexity and the identified needs for improved standards will serve as a foundation for introducing basic concepts to be implemented in BMEcat 2.0 (Section 5). Due to its preliminary status and the overall complexity of the BMEcat specification, we will outline basic concepts only. Finally, we will summarize our work and point out some implementation aspects (Section 6).

If we limit literature to research that deals with product modeling issues in interorganizational information systems expressly, we can form three main working areas. The first area deals with new modeling and configuration concepts that take interorganizational requirements into account (e.g., [4]). This work mainly aims at improving knowledge-based algorithms for configuration processes as the core of sell-side application systems (e.g., [5]). Recently, the semantic web and ontologies gave a new impulse (e.g., [6]). This topic is not in our focus, since we do not aim at developing new methodologies, but adopt classic configuration principles like bills-of-material and search for XML-based representations. Another area is built of work in the context of mass customization as a strategy that integrates construction, production and distribution management [7]. Research work on syntactical and semantic aspects of B2B product data exchange forms the third area. It is characterized by domain-specific issues, i.e. exchange protocols for catalog data and reference models for products, price and classification information (e.g., [8]). We address this area mainly.

In addition, the approach in our paper distinguishes from most research work in product data management, because it focuses on the distribution phase of the product life-cycle, not on the early phases (e.g., planning, engineering).

3. PRODUCT MODELS

Product models can be differentiated by their complexity. According to [9], we distinguish fix, parameterizable and configurable products.

3.1 Fix Products

The first complexity level is limited to the description of fix products which do not need to be configured. In a simple case the description of such a product is realized by a continuous text which contains all relevant information. In practice these descriptions are often used to transfer a lot of information in a proprietary structure. The problem is that these texts are not only specifying different product characteristics, but are also important for order processes; for instance, if they contain information about special prices, product availability or minimum order quantities. These specifications can not be interpreted by a catalog application.

The following example shows how product characteristics are often described: "10-60 Nm; 12.5=1/2; 392mm L; acc. DIN ISO 6789 (4.3.2 < 1 sec.); ± 4% Tolerance; right/left; Plastic Knob; Safe-Boxes avail." This description is understandable from an expert's point of view but it can not be assumed that this knowledge is equally distributed among all buying employees. In addition, it is difficult to compare the product with another one from the same product group, because texts have to be compared which might be structured in different ways.

Introducing a set of product attributes describing these products (here: i.e. max. torque, square wrench size, length, quality standard, tolerance) is not the solution to overcome heterogeneity in product description. To ensure a comparable product specification among all products of one product group, standardized sets of attributes have to be followed. These sets of attributes are provided by standardization organizations as part of or in addition to a standardized classification system (e.g., eCl@ss, EGAS, eOTD). A set of attributes defines for one product class which product attributes must be used to describe a product belonging to this class.

3.2 Parameterizable Products

The next level of complexity arises from the fact that single attributes are not sufficient for describing product variants. Product variants are a set of products which can be distinguished by a few attribute values especially when these values are selected from a predefined list. If product variants are represented by fix products, all possible combinations of attributes and values must be defined as individual products. In fact of the non-linear increasing number of possible combinations, a small amount of attributes already leads to a considerable number of products with an almost identical and therefore redundant description. Furthermore, the connection between the variants is lost for the user of a catalog application. The solution for this problem is to define in addition to static attributes so called variable attributes and allowed values for these attributes.

In order to identify a product, e.g. for the order process, it is necessary to select the base product and to fill all variable attributes with values. Eventually an order number is built by combining the product identification number of the base product with the coded values of all variant attributes. If product variants are described only by a selection of attribute values the order number can be built by concatenating the base product number with the attribute value codes.

The generation of a valid product order number is more difficult. In these cases the following principle is used. The selection or input of attribute values is determining further attributes – in addition to the derived order number. Thus a dependency between the specification of non-fix attributes and other elements of the product description exists. These dependencies have to be considered. But this is sometimes difficult or not possible at all. For example, if a product is identified by an EAN (European Article Number) then it is difficult to map a single product specification including non-fix attributes to a set of EAN, because each EAN is assigned in an independent manner (of course within the supplier's EAN domain), and thus follow no formula. Further examples for variant-dependent product data are figures, description texts, delivery time, availability, and especially the product price.

3.3 Configurable Products

So far we have discussed products only that were specified by discrete attributes; though in practice product configuration is not characterized by a close relation between a product feature and a single attribute but by the necessity to select from one or more components (device, assembly). These components are products in their own that can be described by the same set of data structures (price and order information, static and variable attributes, configuration). However, it is necessary to determine whether the component behaves like an ordinary product that can be ordered independently from a configuration process. The role that a product plays in configuration processes is described by semantic relationships between products (very similar to the bill-of-material concept).

Interdependencies exist between selected components or even between attributes and components similar to relationships between attributes (values of attributes respectively). These dependencies can be very complex; they require a flexible rule-based modeling concept (constraints). For example, selecting the rechargeable battery (component) for a cell phone (base product) determines the speech/stand-by time as well as the weight of the device (attributes).

Assigning values to variable attributes and selecting components in nearly all cases has an effect on the product price. Besides a completely independent price specification (defined price for each variant or configuration), often a flexible system of allowances and charges to the basic price is used. This information extends the bill-of-material.

4. PRODUCT CONFIGURATION IN E-PROCUREMENT AND E-SALES

Concerning e-procurement and e-sales, we have to consider current practice in product data exchange between suppliers and buyers. One alternative to buy-side systems containing all supplier product data is integrating both systems by transferring synchronous messages. Hence, the product model and configuration problem is solved by relying on proprietary, sell-side implementations.

4.1 Sell-Side Configuration

When coupling sell-side and buy-side systems, only a small part of all the product data is transferred via a product catalogs from the supplier to the buyer. This data forms the basis so that the products are findable by search and navigation mechanisms, and posses a meaningful description consisting of product name, basic attributes, keywords and so on. Additionally, the respective product, group or class has an URL (uniform resource locator) pointing to the sell-side system of the supplier. If the buyer selects such a product, group or class in his e-procurement system, a remote catalog access is started: The procurement process is carried forward to the remote sell-side system. In this system, the buyer can select products, specify variants or configurable products; in other words he fills a shopping cart in the remote systems. When he

ends the remote session the shopping cart containing the necessary order information (product identification, unit, price, quantity, delivery time, and additional specification) is returned to the e-procurement system. There, the content of this shopping cart is merged with an existent or converted into a new shopping cart. The buy-side procurement process can be continued.

One advantage of the remote catalog access concept is that even complex configurations on the basis of expert systems and with direct integration with the supplier's ERP system can be realized, without the need of transferring all the product information within the catalog. A powerful product model is not needed. By this approach, a catalog-creating company can bypass the creation and update of extensive catalog data and prevent that valuable product knowledge is transferred to customers, or even competitors. Additionally, the connection to ERP systems enables the calculation of real-time availability and price information.

The application area of the remote catalog access concept is not limited to complex products and connecting sell-side systems. Especially large or constantly changing assortments of standardized goods are suitable. For example, it is not reasonable that a purchasing company builds up and maintains a catalog for books and magazines. In this case it would be advantageous to establish a remote catalog access to the sell-side system of a service provider who is a specialist for the whole assortment of books and magazines. Another scenario is to establish a remote access to marketplaces which offer a high number of supplier catalogs.

There are some disadvantages and limitations in using the remote catalog access. When switching to an external application the user acts in a new environment which differs from the original catalog application, in both handling and functionality. The integration of the remote sell-side application (or marketplace system) and the in-house purchasing organization is difficult and sumptuous. On one hand, established workflows, authorization and budget constraints are bypassed, on the other hand it can not be guaranteed that the product prices coming from the remote system are compliant with bilateral agreements between buyer and seller. Additionally, there is a danger that buyers will order products which are not approved because the buying company has no control over the assortment of goods in the remote system.

4.2 Capabilities of XML-based Standards

The standards for electronic product catalogs in e-procurement scenarios do not match the requirements on the representation of parameterizable or configurable products. Many catalog standards like RosettaNet, CIDX or EAN.UCC do not provide any support for complex products, but are designed for describing fix products only. Other standards like BMEcat 1.2, catXML, cXML, eCX, OAGIS or xCBL provide some concepts towards configuration, however, relevant requirements are not covered; thus parameterizable or configurable products can not be handled in a standardized way [3].

Table 1 gives an overview of the modeling concepts that are already covered by catalog standards (for details see [1]).

	BMEcat 1.2	cXML 1.2.008	OAGIS 8.0	xCBL 4.0
Variable Attributes				
Selection of discrete values (Variants)	yes	no	no	no
Cardinality	no	no	no	no
Default Values	no	no	no	no
Input Parameters	no	no	no	no
Relationships: derived Attributes	no	no	no	no
Relationships: derived Parameters	no	no	no	no
Relationships: Product Price	no	no	no	no
Relationships: Order Number	yes	no	no	no
Relationships: Restrictions	no	no	no	no
Selection of Components				
Optional Choice (Product References)	yes	no	yes	yes
Selection Types /Number of Types	yes/9	no	yes/1	yes/4
Selection: Mandatory	yes	no	no	yes
Cardinality	no	no	no	no
Default Values	no	no	no	no
Relationships: derived Attributes	no	no	no	no
Relationships: derived Parameters	no	no	no	no
Relationships: Product Price	no	no	no	no
Relationships: Order Number	no	no	no	no
Relationships: Restrictions	no	no	no	no
Remote Catalog Access	no	yes	no	no

Table 1. Comparison of Product Models in XML Catalog Standards

The empirical study shows that all standards allow a product description based on free-defined attributes. BMEcat is the only standard that supports the description of variants by defining how a variable attribute must be assigned with a value from a domain of discrete values. But the capabilities in BMEcat 1.2 are very limited, primarily because of missing constructs to declare dependencies between attribute values and prices. Furthermore it is not possible to limit the range of permitted variants or to exclude specific variants. Thus its practical usability is questionable. None of the standards contains data structures that represent (price) dependencies, constraints and rules between a basic product and its components; though these concepts are a prerequisite for modeling configurable products.

5. BMECAT 2.0

This Section presents a preliminary version of the BMEcat 2.0 product model. The product model was developed by a working group, consisting of all partners relevant to B2B e-commerce, such as buying companies, suppliers from industry, intermediaries (marketplaces and catalog service providers) as well as software companies specialized in e-procurement applications. All companies are located in Germany (e.g., Deutsche Telekom, cc-chemplorer, Heiler Software). This group compiled, evaluated and committed to a set of requirements. Supplemented by research studies and comparative analysis of competitive standards, the group's technology experts began to specify new and enhanced XML data structures by using XML Schema, the standard schema language for XML exchange formats.

All data models will be visualized by proprietary graphical representations, which are part of XMLSpy, a software tool for XML data modeling. The following table shows the main symbols and gives a short explanation.
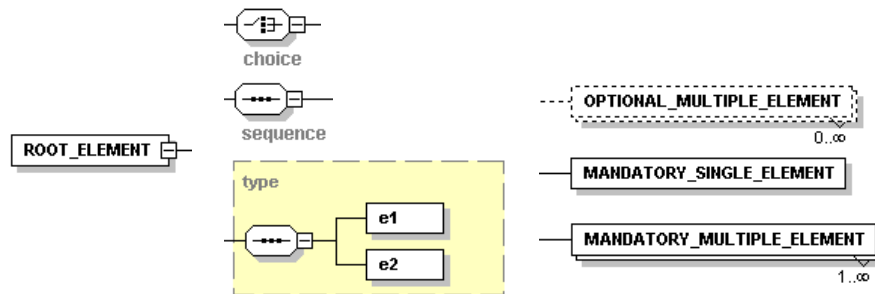
Table 2. Symbols used in XML Schema Diagrams

5.1 Basic Concepts

When modeling configuration, we must adhere to some concepts of product representation in electronic catalogs. These concepts are not specific to configuration, but can be reused and enhanced for configuration purposes. We will describe three basic concepts. These are standardized product classification, price calculation based on formulas, and uniform integration of all products into one catalog.

5.1.1 Standardized Product Classification

Contrary to sell-side systems or proprietary, often CD-ROM based product catalogs (and configurators), a standardized product classification is essential to e-procurement and marketplace systems. Standardized means that all product classes and associated sets of attributes are subject of a horizontal or vertical standard for product classification; hence supplier-specific class hierarchies and attributes are not sufficient. The reason is that catalogs of different suppliers have to be integrated into a multi-supplier catalog. This enables efficient product search, especially attribute-based search, and qualified product evaluation by comparing product properties expressed by attributes and their values. Meeting these requirements, buying companies can establish a common understanding of a product domain, its terminology and semantics.

Modeling electronic product catalogs calls for giving references to product classification systems, product classes and respective attributes whenever a product is categorized and described by attributes. This requirement is fulfilled by the element CLASSIFICATION_REFERENCE as part of product master data. In addition, it ensures that a product can be belong to more than one classification system, but may belong to one class of each system only. Due to the BMEcat 1.2 terminology, attributes are called features.
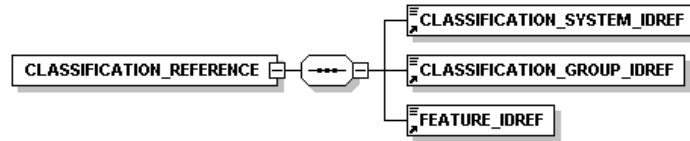
Figure 1. Data Model for Classifying Products.

5.1.2 Price Formulas

From the e-procurement point of view, product catalogs serve as an instrument for getting product information, choosing the right products and starting order processes. A prerequisite for such a decision and process is to get valid price information for each required product. Since prices of configurable products are dependent on selected components and attribute values, the underlying price model must be able to represent the relationships between configuration, attributes and the price. The BMEcat 1.2 price model is quite capable (see [5]), but not sufficient for configuration purposes. Therefore, we extended the price model and introduced the generic concepts of price formulas for representing price information and the described relationships. However, this model is not only used for configuration. It is also suitable for fix products, when the price is dependent on external price parameters. Prime examples are products, which consist of metals or include a relevant amount of metal. Since the metal stock market is quite volatile, the product prices are also dependent on the current market price.

The model for price formulas is subdivided in the definition of formulas (catalog header) and the application of formulas on the product level. Therefore, formulas need to be defined only once, and can be reused for different products of the same catalog.

The element FORMULAS as part of the catalog header builds a repository of formulas (see figure 2). FORMULA_ID and FORMULA_VERSION indentify each formula. The formulas are described by the elements FORMULA_NAME and FORMULA_DESCR; the element MIME serves for multimedia objects (files) relevant to formulas (e.g., drawings, specifications, standards). FORMULA_SOURCE gives a reference to the organization or company that defines and maintains the formula. This enables to build industry-wide repositories. In course of the BMEcat 2.0 development process, a couple of industry associations have already proposed their intention to standardize compliant formulas for their domains. These will be mandatory for all respective suppliers.
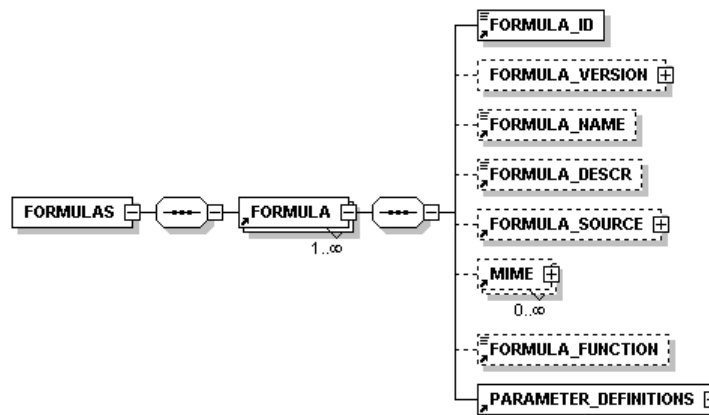
Figure 2. Data Model for Price Formulas.

The actual definition is subject of the element FORMULA_FUNCTION and a list of parameters that are part of the function (see figure 3). This fulfills two needs. First, it is possible to calculate the product price by using the formula, its parameters and respective parameter values. Second, the parameter can be described is such way, that the price calculation is meaningful, understandable, and traceable. In some branches of industry, defined parameters are already a common good that the purchasing decision depends mainly on the parameter values; calculating the exact price is not necessary.
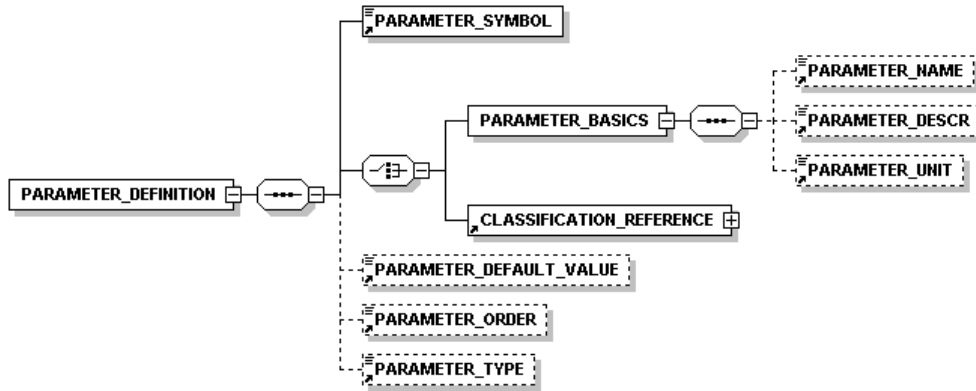
Figure 3. Data Model for Parameter Definition.

The syntax of FORMULA_FUNCTION is equal to functions or expressions in JavaScript, thus it is standardized, not proprietary. Speaking of PARAMETER_DEFINTION, we added CLASSIFICATION_REFERENCE, so that attributes of classification systems can serve as parameters, too (e.g., length, diameter, weight).

The element PARAMETER_DEFAULT_VALUE holds default values for parameters. In many cases, these values will be overwritten on the product level. Since formulas are applied during the run-time of a catalog different data sources can be used to get the actual values. We postulate three categories of parameters (PARAMETER_TYPE): The first one covers parameters which have a given, static value that is already attached during the build-time; these are not real parameters. The second category consists of parameters, whose values can be accessed from a web resource (e.g., URI, web service of a metal stock market). The third category groups parameters that get their values only by user interaction or requesting data from an internal software system.

Price formulas are applied on the product level by giving a reference to their identifiers (FORMULA_IDREF, see Figure 4). In many cases, parameter values will be overwritten or instantiated with product-specific values; this is the role of the PARAMETER element. PARAMETER_SYMBOLREF names the parameter, PARAMETER_VALUES sets the product-specific value. However, default values can be used directly, too.
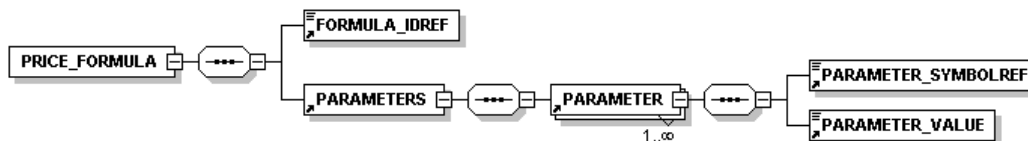
Figure 4. Data Model for Price Formulas on the Product Level.

5.1.3 Uniform Integration of all Products into one Catalog

One of the main goals of the BMEcat initiative is to transfer products of all kinds of complexity in one electronic product catalog. In today's e-procurement systems, due to different catalog exchange formats there are often separated electronic product catalogs for fix products and for products which are handled by remote catalog access. Therefore users of e-procurement systems have to know whether the product they are searching for is configurable or not in order to use the right catalog. Additionally the possibilities on describing the products differ very much between 'fix' and 'remote access' catalogs. Product catalog standards for fix products in most cases offer a wide range of support for the selection process like keywords, attributes, pictures and product classification whereas standards specifying the remote catalog access only provide very limited description possibilities [1].

The approach of BMEcat is to integrate products of all degrees of complexity in one catalog and providing the full set of description options. This allows using the same search and navigation principles like parametric search, keyword search and hierarchical navigation using a class tree. Another advantage is that the catalog creator is able to present his whole assortment of goods in a uniform way. Not until the start of the ordering process there have to be different procedures according the degree of complexity of the products. So it is possible to integrate all products which may vary from fix products, via simple configurations to complex products processed by means of the remote catalog access.
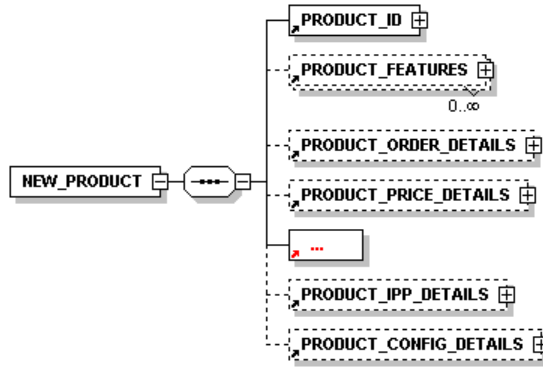
Figure 5. Data Model for Products.

5.2 Configurations

Configurations are modeled in BMEcat by the element PRODUCT_CONFIG_DETAILS. This element is used to represent parameterizable and configurable products. Only the product model, this means the structure of the product is specified here. There are no specifications how the configuration process has to be carried out, or how the GUI representation has to look like. These details are subject of the target software system. It was one of the main goals to build a product model which enables the catalog creators to extend the fix products with simple configurations in an easy way. To establish this, BMEcat forbear from some more complex modeling options like the definition of rule based settings of values. For products which are too complex to represent in BMEcat 2.0, there is still the alternative to use the remote catalog access.

5.2.1 Stepwise Configuration

The whole configuration process is divided up into single steps. With each of this steps one variant product characteristic is being defined. This product characteristic is specified by setting a value for a variant product feature (CF_FEATURE), or by choosing a sub component (CF_PART) from a list of alternatives. Whether more than one product characteristic is determined at one time (on one page) or step-by-step with only on product characteristic each step (one step on each page) depends on the implementation of the target system.
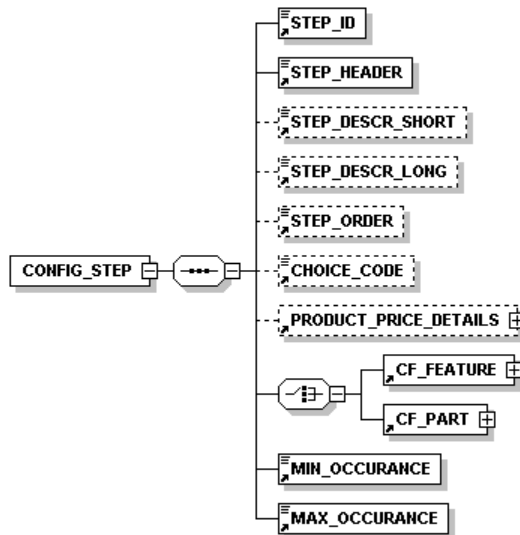
Figure 6. Data Model for Configuration Steps.

STEP_HEADER, _DESCR_SHORT und _DESCR_LONG are used for a textual description of the configuration step. With the element STEP_ORDER the sequence of the configuration steps can be specified.

106

The element CHOICE_CODE offers the possibility to specify a code which can be used at the end of the whole configuration process to generate an unambiguous order number.

To specify a basic price for one configuration step the element PRODUCT_PRICE_DETAILS is used. This basic price will be entered in the calculation of the total price of the fully configured product. In this context, the above introduced price formulas are used. Therefore it is possible to compute variant prices which are depending on the already completed configuration steps.

Due to the fact that total price and order number can also be computed by the configuration rules both fields are optional elements.

The structure of the product is represented by the elements MIN_ and MAX_OCCURENCE as well as CF_FEATURE and CF_PART. The set of cardinalities which are represented in the OCCURRENCE elements are defining how often a product characteristic can be and has to be specified by the user.

The element CF_FEATURE is used for the definition of product characteristics which can be specified by setting a value of a parameter or by selecting one or more values from a list. By means of the element CF_PART sub components, which can be full products itself, might be chosen from a list. Most of the time it is possible to specify the same product characteristic either as parameter or as sub component. Which of these two alternatives is chosen by the catalog creators depends on how detailed the description of the product characteristic has to be.

5.2.2 Parameterizable Product Attributes

To specify a parameterizable product characteristic either an attribute, which is already defined within a classification system, can be used or a new attribute can be defined by the element FEATURE_DEFINITION. Additionally, a file with a picture or a data sheet can be attached to the catalog using the element MIME to provide a more detailed description of the attribute and its values.
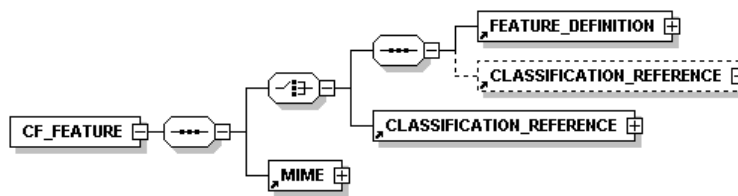
Figure 7. Data Model for Parameterizable Attributes.

The element FEATURE_DEFINITION allows making a detailed definition of all information relevant for the attribute, in particular the domain of the values. This element is used in BMEcat also for the definition of attributes in the context of classification systems and for fix products. The domain of the values can be a basic data type like float or string, but can also be an enumeration. For every value of this enumeration list an additional CHOICE_CODE and individual price information can be provided, which are also included in the final price calculation and generation of the order number.
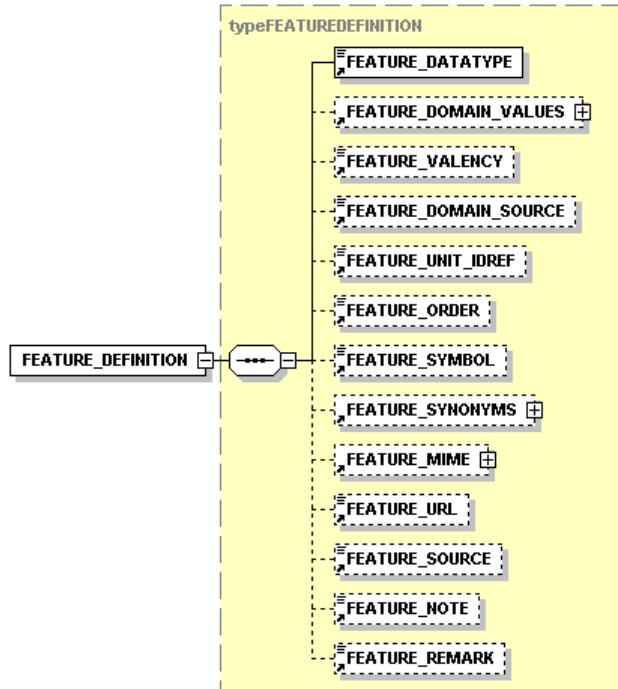
Figure 8. Data Model for Attributes.

5.2.3 Component-based Product Characteristics

Some product characteristics are specified by selecting components from a given list of parts (element PRODUCT_CHOICE). This information is stored in element CF_PART.
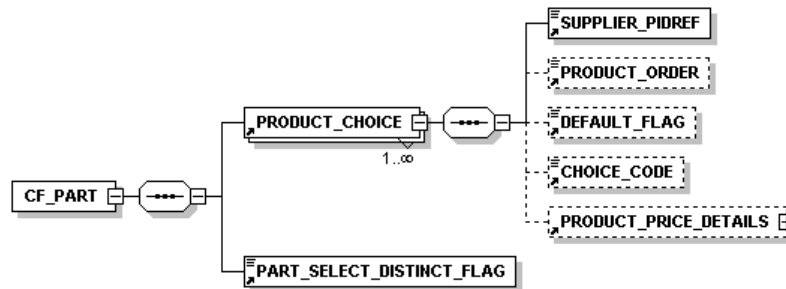
Figure 9. Data Model for Components.

Suppliers can build a list of alternative parts and decide whether different parts must be selected if it is allowed to select more than one part. Each component is referenced by using the supplier-specific product ID (SUPPLIER_PIDREF). The advantage is that such a component can be described by all data structures available for products, i.e. attributes, keywords, descriptions, multimedia object etc. However, we have to consider that not all components can be ordered like ordinary products; this is marked by a flag. If a component is described like a complete, single product, this component may be configurable as well (recursion). This allows to build a hierarchy (bill-of-material) that has to be followed when configuring a product. The sequence of parts having the same father component can be controlled, too (PRODUCT_ORDER).

5.2.4 Configuration Rules

Assuring that each customized configuration results in a valid configuration requires a set of rules. These conceptualizations of configuration knowledge can be classified to (a) rule-based, (b) model-based and (c) case-based

approaches [10]. Configuration rules fulfill two roles. First, they assure consistency. Second, they lead to modifications of product attribute values.

The BMEcat 2.0 product model can be classified into the model-based approaches, especially constraint-based approaches. Constraints among components restrict the ways components can be combined. The syntax is restricted to the minimum; it allows only the definition of valid and invalid combinations of product characteristics.

Concerning the second role, all attributes are subject of constraints, thus they can be modified. The specification is described in a rule-based manner leading to if-condition-then-modification statements. All product attributes can be referenced on the right side of these expressions (e.g., if component = "wheel" then total_weight = total_weight + wheel.weight). This covers also order number and product price of the configuration.

The drawbacks of combining rule-based and constraint-based approaches are counter-balanced by advantages for catalog creators, who can easily describe configuration rules (due to rather the simple syntax).

6. CONCLUSIONS

In this paper we have presented a preliminary version of BMEcat 2.0 for complex products in e-procurement and e-sales. First, we distinguished different levels of complexity. Then we evaluated the current state of XML-based exchange format for product catalogs briefly. Eventually, we introduced basic concepts of BMEcat 2.0 and described their application and benefits to widening the scope of e-procurement on the buy-side.

While our approach is driven by industry requirements, we aimed at developing conceptual models that will last for a longer time. One technical obstacle can be the complexity of the exchange format itself, that might be slow down the proliferation. However, we believe that our model is quite balanced. Time will tell if software companies will come up with first implementations in 2004.

6. REFERENCES

1. Leukel, J., Schmitz, V. and Dorloff, F.-D. (2003). B2B E-Procurement Beyond MRO? Proceedings of the 6th International Conference on Electronic Commerce Research (ICECR-6), pp. 493-500.

2. Schmitz, V., Kelkar, O. and Pastoors, T. (2001). Specification BMEcat, Version 1.2. URL: http://www.bmecat.org.

3. CEN/ISSS (2004). WS/eCat Multilingual catalogue strategies for eCommerce and eBusiness. Preliminary Workshop Agreement, Brussels, Belgium.

4. Møller, J., H. Reif and Hulgaard, A. H. (2001). Product Configuration over the Internet. Proceedings of the 6th INFORMS Conference on Information Systems and Technology (CIST 2001).

5. Hadjhamou, K., Caillaud, E. and Lamothe, J. et al. (2001). Knowledge for product configuration. Proceedings of the International Conference on Engineering Design (ICED'2001), pp 131-138.

6. Fu, X., Li, S. and Chen, G. (2002). Semantic Web Based Distributed Configuration. Proceedings of the International Workshop on Grid and Cooperative Computing (GCC2002), pp. 389-401.

7. Rautenstrauch, C., Seelmann-Eggebert, R. and Turowski, K. (Editors)(2002). Moving into Mass Customization - Information Systems and Management Principles. Springer, Berlin et al., Germany.

8. Kelkar, O., Leukel, J. and Schmitz, V. (2002). Price Modeling in Standards for Electronic Product Catalogs Based on XML. Proceedings of the 11th International World Wide Web Conference (WWW2002), pp. 366-375.

9. Stahl, A., Bergmann, R. and Schmitt, S. (2000). A Customization Approach for Structured Products in Electronic Shops. Proceedings of the 13th Bled Electronic Commerce Conference, pp. 252-264.

10. Sabin, D., and Weigel, R. (1998). Product Configuration Frameworks – A Survey. IEEE intelligent systems, 13, pp. 42-49.

APPLYING FUNCTION POINT ANALYSIS TO EFFORT ESTIMATION IN CONFIGURATOR DEVELOPMENT

A. Felfernig, A. Salbrechter

Department of Business Informatics and Application Systems
University Klagenfurt
Universitätsstraße 65-67
A-9020 Klagenfurt, Austria
Email: {felfernig, salbrechter}@ifit.uni-klu.ac.at

Knowledge-based configuration is a successful application of Artificial Intelligence approaches in industry (e.g. telecommunication industry, financial services, or automotive industry). The increasing complexity of configurable products and services necessitated improved expressiveness and maintainability of knowledge representation languages empowering the development and maintenance of large and complex configuration knowledge bases. Within the context of such configuration projects, the effective integration of effort estimation techniques considering the peculiarities of configuration system development is still an open issue. We discuss the application of Function Point Analysis (FPA) (a standard Software Engineering method for determining implementation efforts) in the context of knowledge-based configuration projects and present a framework for a company-specific implementation.

Significance: extension of the scope of Software Engineering effort estimation approaches to the development of knowledge-based configuration systems.

Keywords: knowledge-based configuration, project management, effort estimation.

1. INTRODUCTION

Configuration can be seen as a special kind of design activity [12], where the final product is built of a predefined set of component types and attributes, which can be composed conforming to a set of corresponding constraints. Configuration systems are of strategic importance for enterprises dealing with highly variant products and services, e.g. response and delivery times to the customer are reduced and invalid orders can be prevented by automatically checking the customer requirements w.r.t. given marketing constraints, technical constraints and constraints related to production processes.

Since the development of the product and the product configurator has to be done concurrently, configurator development- and maintenance time is strictly limited, i.e. the implementation of configuration systems is a critical task and organizations dealing with the provision of highly variant products and services recognize the importance of available measures for analysing the efforts associated with the development and maintenance of configuration systems. Effort estimation is a crucial factor when determining the feasibility of a project, creating an offer, or managing resources. As a rule, configuration systems are not standalone systems but have to be integrated into already existing software environments. In this context project managers implementing configuration applications should not be forced to apply additional effort estimation methods but rather be instructed how to effectively apply conventional Software Engineering approaches to knowledge-based systems development. However, within the context of configuration projects, the effective integration of effort estimation techniques considering the peculiarities of configurator development is still an open issue.

In this paper we sketch how Function Point Analysis (FPA) [1,2] can be applied to effort estimation in knowledge-based configuration systems development. FPA is based on a user- (requirements-) centered view on the software and is platform-independent. The method has first been proposed by [2] with the goal to provide an effort measure for the functional size of software - together with the counting rules it has been adapted several times. Currently, it is maintained by the International Function Point Users Group (IFPUG). Using FPA we can determine the implementation efforts related to a project (development of knowledge base, development of user interface, development of interfaces to remote applications, e.g. product catalogs or online sales applications). Applying FPA to configuration software development extends the scope of Software Engineering estimation approaches to knowledge-based system development. Thus knowledge-based systems development is made transparent within industrial software development processes and effort estimation for traditional software development projects is integrated with effort estimation for knowledge-based software development projects.

The remainder of the paper is organized as follows. In Section 2 we discuss and exemplify basic principles of knowledge-based configuration. In Section 3 we present an effort estimation process applicable to configuration system development. In Section 4 we show under which conditions FPA can be applied to effort estimation in configurator development. In Section 5 we discuss issues related to the application of the presented concepts. Section 6 contains related work.

2. CONFIGURATION KNOWLEDGE REPRESENTATION

As pointed out in [16] the modeling of configuration knowledge is a critical task - any framework must address the issues of expressiveness and representational power and provide mechanisms for coping with the high rate at which knowledge changes. In many cases the used description languages for building configuration knowledge bases are not integrated into industrial software development processes. These description languages are difficult to communicate to domain experts which makes it demanding for software development departments to incorporate such technologies into their standard development process. For the realization of configuration systems the Unified Modeling Language (UML, [11]) can be used as notation in order to simplify the construction of a configuration knowledge base [6]. The usage of UML for configuration knowledge representation makes sense for the following reasons:

- UML is widely applied as standard design language in industrial software development.

- UML is extensible for domain-specific purposes, i.e. (using profiles) the semantics of the basic modeling concepts can be further refined in order to be able to provide domain-specific modeling concepts (e.g. modeling concepts for the configuration domain).

- UML has a built-in constraint language (the Object Constraint Language (OCL) [15]). UML and OCL are the perfect combination of representation concepts for designing configuration applications.

In the following the simple UML configuration model of Figure 1 will serve as working example in order to show the application of the presented effort estimation concepts[5].
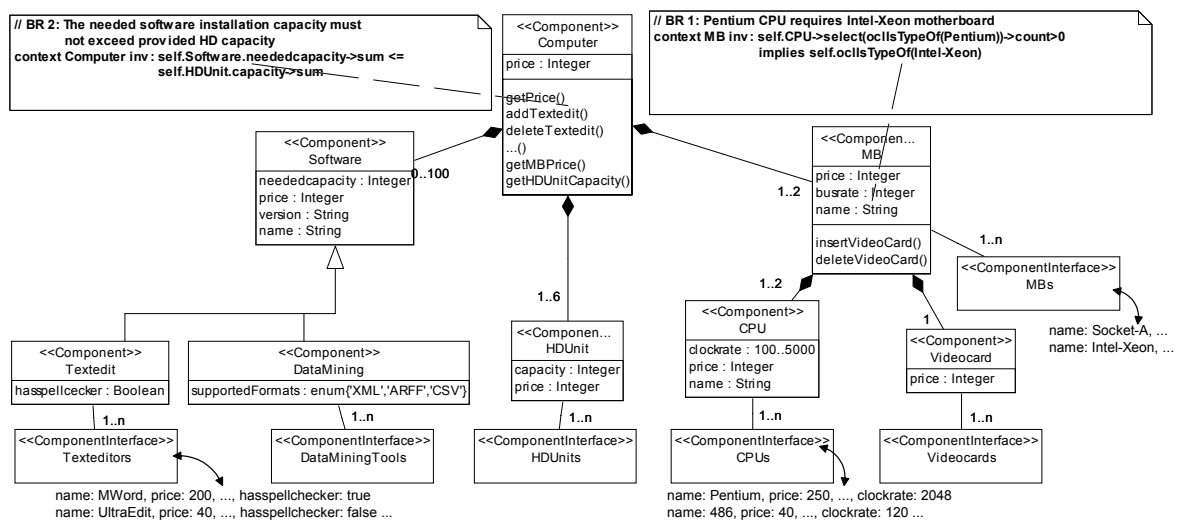
Figure 15: Example configuration model

This model represents the generic product structure, i.e. all possible variants of a configurable computer. The basic structure of the product is modeled using component types (basic building blocks the final product can be built of), generalization hierarchies, aggregations and interfaces to different product catalogs[6]. The set of possible products is restricted through a set of business rules (BR1, BR2 in Figure 1) related to technical restrictions, economic factors and restrictions according to the production process. Such a generic description of a product structure can also be denoted as domain description (DD) [7]. The used modeling concepts are defined in a UML configuration profile [6] and can be

---

[5] Note that the completion of the UML configuration model is not a precondition for applying the presented estimation concepts. However, the more is known about the product structure and functions, the more exact are the estimates.
[6] Note that not all product catalog instances (related to component interfaces) are shown here completely.

interpreted as ontology [5], i.e. ontologies are theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge[7].

Most configuration tasks incorporate additional restrictions (e.g. customer requirements) defining components or attribute settings which must be part of the final configuration. These requirements are called systems requirements specification (SRS) [7]. An example for such customer requirements is the following: set the maximum overall price of the configuration to 1000. A configuration result calculated by a configuration system (configurator) can be interpreted as an instantiation of the configuration model, where all business rules and customer requirements are satisfied. A configuration result can be represented as UML instance diagram [6].

3. EFFORT ESTIMATION PROCESS

There exists a number of approaches investigating the application of Function Point Analysis (FPA) for object-oriented software development (e.g. [8,14]). However, a direct application to effort estimation in configuration software development results in significant deviations. The reasons for these deviations are the following:

- Knowledge-based systems development: existing approaches to FPA (see [1]) do not provide a standard way of accounting for the size of certain types of functional user requirements, notably complex sequences of rules as found in knowledge-based systems [1]. Such mechanisms are needed in order to support reasonable effort estimates for knowledge base development and maintenance.

- Adjustment factors: important adjustment factors (e.g. experience of project members) currently not included in the FPA have to be introduced within the context of knowledge-based configuration. Furthermore, statistical spread resulting from the analysis of empirical data can exceed the standard deviation of FPA adjustment factors, i.e. the calculation of adjusted function points has to be adapted.

- Company-specific software process: FPA doesn't consider company-specific properties (e.g. time spent by programmers to complete a specific implementation task) of the development process but rather uses a standard complexity estimation to predict efforts. This results in deviations of estimates from actual project implementation efforts. Company-specific complexity measures can help to improve the accuracy of effort estimates.

The basic process for company-specific Function Point analysis is depicted in Figure 2. In general it is very difficult to collect correct and useful software process data (e.g. the time spent to implement a specific class method), because software developers do not consider process data collection as an important activity (compared to concrete coding tasks). Software projects often lag behind defined deadlines – manually collecting data is a time-expensive task, at the same time there is no time left to be spent for data collection. The best solution for this problem is an automated acquisition tool seamlessly integrated into a developers implementation environment, i.e. developers are not required to interrupt their main tasks.
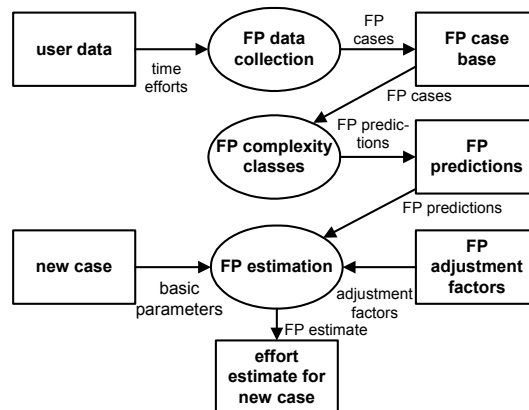
Figure 16: Effort estimation process

In the phase *FP data collection* data collection can be performed by such a tool [13] – the result is a set of measured values describing time efforts related to one concrete project. Based on the result of this phase, estimates for different complexity

---

[7] For a detailed discussion on the modeling elements of a configuration ontology and their translation into an executable representation see e.g. [6].

classes can be derived by applying cluster analysis on the given case base (determination of the boundary values for the complexity tables discussed in Section 4 – done in the phase *FP complexity classes*). The resulting tables change over time, i.e. improved development performance is automatically propagated to the corresponding complexity tables. Finally, given a new case (new project), effort estimates for the new project can be calculated (phase *FP estimation*). The more detailed the specification for the new configuration system is, the more exact are the corresponding effort estimates, i.e. we can start with a few basic assumptions on the number of classes, attributes, methods and business rules and will receive a first rough estimate. Throughout the project the estimates become increasingly exact.

4. EFFORT ESTIMATION FOR IMPLEMENTING CONFIGURATORS

4.1 Approach to FPA

Our approach to FPA in knowledge-based configurator development as well assumes a user-centered view on a system. The functionality of the configurator application is defined by the following factors (see Figure 3).
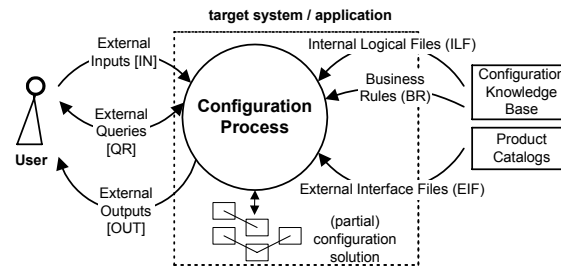
Figure 17: FPA areas

1. EI - External Input, i.e. those SRS[8] related to functions which change the actual configuration setting, e.g. inserting a new software component or limiting the maximum price of the overall configuration. Using EI functions a user can add, change and delete settings of a configuration.

2. EQ - External Query, i.e. those SRS related to functions displaying specific data from the current configuration setting, e.g. the price of a certain CPU part of the actual configuration. External Query (EQ) functions allow users to select and display specific data from configuration settings. For this purpose the user enters selection criteria which are used to match with configuration data, i.e. no data manipulation but a direct retrieval is performed by External Queries.

3. EO - External Output, i.e. those SRS related to functions generating output for the user (generation is based on calculations), e.g. the determination of the minimum hard-disk capacity needed for the installation of a certain text editing environment or the inclusion of a pension calculator in the investment portfolio configuration process.

4. ILF - Internal Logical File. EIs, EQs, and EOs operate on a model of the domain description (DD). In terms of FPA, the configuration knowledge base is denoted as a set of Internal Logical Files (ILFs), i.e. knowledge elements which are maintained[9] within a configuration application. ILFs allow users to utilize data they are responsible for maintaining.

5. EIF - External Interface File. Product catalogs can be seen as an example for External Interface Files (EIF), i.e. knowledge elements which are maintained outside the configuration application. EIFs allow users to utilize data they are not responsible for maintaining (e.g. product data from an external Enterprise Resource Planning system).

6. BR - Business Rule. Conventional FPA approaches [1] do not explicitly consider the complexity of business logic - configurators are knowledge-based applications where knowledge complexity has a strong influence on development time and costs. In order to consider this important aspect in our estimation approach, we introduce Business Rules as an additional complexity dimension.

---

[8] In this context, SRS are not strictly interpreted in the sense of [7], i.e. are more related to functions or methods which impose requirements to the configuration system. Examples for SRS are e.g. adding a certain Videocard or changing the value of the CPU clockrate to 2Ghz, where the semantics is that the added Videocard must be part of the configuration and the clockrate of a certain CPU must be set to 2Ghz.

[9] In this context maintenance denotes the usage of user functions for adapting the current model (i.e. instance of the configuration model) conforming to the given user requirements.

In the following EI, EQ, and EO are denoted as *transactional function types*, ILF, EIF and BR are denoted as *data function types*. Based on the following three tasks, a function point value can be calculated for a given project.

1. The components *External inputs* (EI), *External Outputs* (EO), *External Queries* (EQ), *Internal Logical Files* (ILF) and *External Interface Files* (EIF) are identified. ILFs, EIFs, BRs, EIs, EOs and EQs can be directly identified from a given UML configuration model - rules for identifying those units are discussed in Section 4.1 and 4.3. In early project phases assumptions on the number of classes etc. serve as an input for the first approximate estimates.

2. The complexity weights are assigned to each of those components using the levels *low*, *average* and *high*. For each data function, *Record Element Types* (RETs) and *Data Element Types* (DETs) are counted as basic parameters. Based on those parameters the complexity of each data function can be determined. For each transactional function, *File Types Referenced* (FTRs[10]) and *Data Element Types* (DETs) are counted as basic parameters for determining the complexity (low, average, high) of the transactional function. By applying Tables 1-4 the complexity of each data function and each transactional function can be determined. The application of Table 5 results in a value for unadjusted function points (UFPs) for the configuration application, i.e. UFP = $\sum EI_{FPs} + \sum EO_{FPs} + \sum EQ_{FPs} + \sum BR_{FPs} + \sum ILF_{FPs} + \sum EIF_{FPs}$.

3. Finally, General System Characteristics (GSC) are investigated w.r.t. their influence to the calculated UFPs, i.e. Adjusted Function Points FP = UFP*Adjustment Factor.

4.2 Data collection

As already mentioned, company-specific complexity measures can help to improve the accuracy of effort estimates. Figure 4 sketches a corresponding data collection and transformation process. Developer's efforts to implement transactional- and data function types are stored in a set of interaction traces. Theses traces are analyzed with the goal to derive the corresponding boundaries and measures for the complexity tables 1-4. Our approach to determine the boundaries is to generate a fixed set of clusters (data mining) representing the different entries in a complexity table. The values and boundaries in the table change (decrease) as the quality of the development process improves. By aggregating the stored time efforts for a certain complexity class (e.g. Table 1, 1-19 DET x 1 RET) into a corresponding average value, the entries of Table 5 can be determined (the basic assumption is that 10 Function Points represent 1 MM[11]).

The entries of Tables 1-5 represent our current experiences in implementing commercial configuration applications. However, these tables are repeatedly improved in order to be up-to-date with the current development process. The *Deviations* in Figure 4 are the basis for providing boundaries for the calculated effort estimates (i.e. Function Points).

4.3 Data functions

Definition 1: Identification of Logical Files

Logical Files (LFs) can e.g. be identified using the following criteria which are based on a variant of the approach presented in [4]. A logical file (i.e. either an ILF or an EIF) is identified by combining the following two basic rules.

1. Count an entire aggregation structure as a single logical file, recursively joining lower level aggregations.

2. Given an inheritance hierarchy, consider as a different logical file the collection of classes comprised in the entire path from the root superclass to each leaf subclass, i.e. inheritance hierarchies are merged down the leaves of a hierarchy. [1]

Merging superclasses makes sense since leaf classes with all inherited structures are instantiated during a configuration process. Figure 5 (left) contains an abstract example for the application for the above mentioned rules, i.e. LF_1 represents those classes forming a partof hierarchy, LF_2 and LF_3 represent logical files derived from the different paths to leaf subclasses in the generalization hierarchy, where LF_3 also includes the partof relationship between classes B2 and F (combination of rule 1 and rule 2).

Note that this is one of several alternatives for the identification of Logical Files (see [4]). An alternative approach currently investigated is to interpret interaction units (i.e. input masks) as basic units for the identification of (internal) logical files.

---

[10] Referenced Logical Files – see Section 3.3.
[11] See also Section 4.6.

Example 1: Identification of LFs

In the configuration model of Figure 1 the following LFs can be identified:

- ILFs: {Computer, Software, HDUnit, MB, CPU, Videocard}, {Software, Textedit}, {Software, DataMining}

- EIFs: {Texteditors}, {DataMiningTools}, {HDUnits}, {CPUs}, {Videocards}, {MBs} [

The Logical Files identified for the example configuration application are shown in Figure 5 (right).

ILF/EIF	StartTime	EndeTime	...
ILF1	10:45	10:56	...
EIF1	09:10	09:22	...
ILF2	13:22	13:28	...
...	...	...	...

Business Rule	StartTime	EndeTime	...
Business Rule 1	09:15	09:20	...
Business Rule 1	14:36	14:44	...
Business Rule 2	16:27	16:54	...
...	...	...	...

EI/EO/EQ	StartTime	EndeTime	...
EI1	10:45	10:56	...
EI2	09:10	09:22	...
EI1	13:22	13:28	...
EO1	10:45	10:56	
EO2	09:10	09:22	
EQ1	13:22	13:28	
...	...	...	...

FP complexity classes

Tables 1-4   Table 5   Deviations
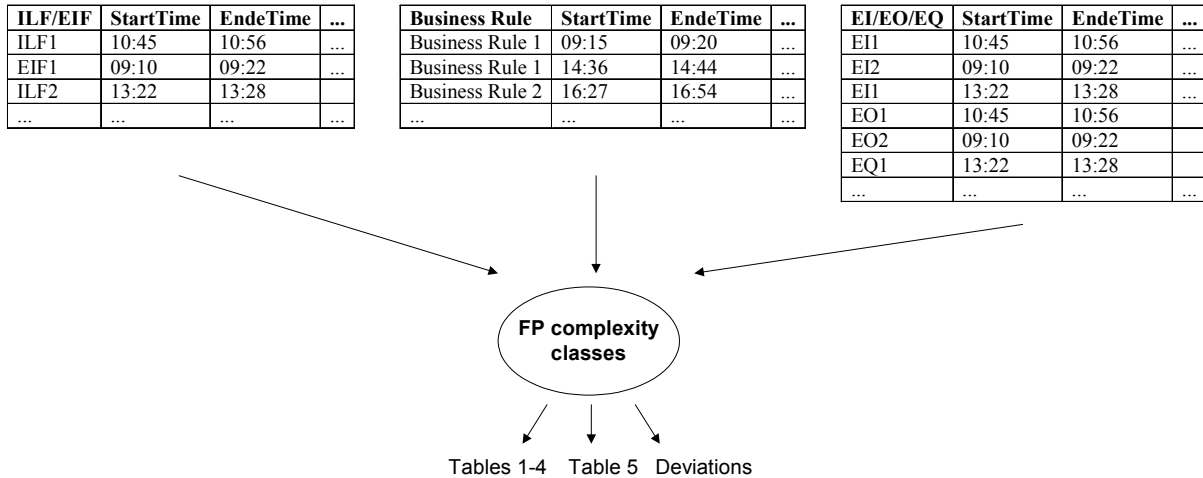
Figure 18: Data collection and transformation process

4.4 Complexity of Data functions

Definition 2: Complexity of LFs

For each LF (ILF and EIF) the number of Data Element Types (DETs - unique user-recognizable fields of LFs) and the number of Record Element Types (RETs - user-recognizable and logically related data as subgroups of LFs) is computed.

1. Each class within a LF is interpreted as 1 RET.

2. Each attribute within a LF is interpreted as 1 DET.

3. Each involvement of a class in an association with multiplicity>1 is interpreted as 1 DET within a LF.

4. Each discriminator to a subclass in a generalization hierarchy within a LF is interpreted as 1 DET. [

Depending on the number of RETs and DETs the complexity of ILFs and EIFs can be determined (see Table 1).
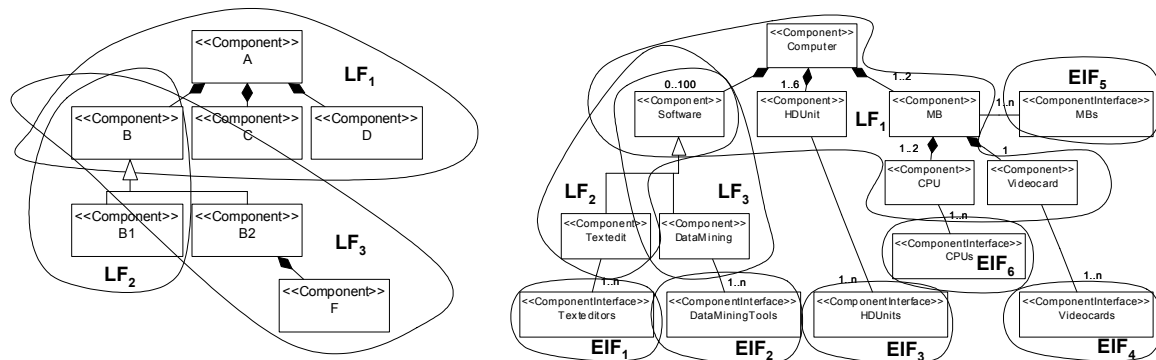
Figure 19: Identification of Logical Files

Example 2: Complexity of ILFs and EIFs

Based on the entries of Table 1 the data complexity of the computer configuration example can be determined as follows[12]:

- ILFs: {{Computer, Software, HDUnit, MB, CPU, Videocard}/[6, 18, average], {Software, Textedit}/[2, 6, low], {Software, DataMining}/[2, 6, low]}.

- EIFs: {{Texteditors}/ [1,6,low], {DataMiningTools}/ [1,6,low], {HDUnits}/ [1,3,low], {CPUs}/ [1,4,low], {Videocards}/ [1,2,low], {MBs}/ [1,4,low]}[13].

	1-19 DET	20-50 DET	> 50 DET
1 RET	low	low	average
2-5 RET	low	average	high
> 5 RET	average	high	high

Table 1: Complexity of Data Functions (DF)

Definition 3: Complexity of BRs

Depending on the number of RETs and DETs referenced by BRs within a LF and the number of BRs related to a LF (BRLF - BRs per LF), the complexity of BRs is determined (see Table 2).

RET+DET	1-4 BRLF	5-9 BRLF	>9BRLF
1-16	low	low	average
17-40	low	average	high
>40	average	high	high

Table 2: Complexity of Business Rules (BR)

This approach provides a measure for the complexity of business rules defined within a LF. Note that this is a basic approach for estimating business rules complexity - if needed, this approach can be refined and extended for the specific settings of a company.

Example 3: Complexity of BRs

Based on the entries of Table 2 the BR complexity of the computer configuration example can be determined as follows[14]:

{{Computer, Software, HDUnit, MB, CPU, Videocard}$_{BR1,2}$/[10,2,low], {Software, Textedit}$_{BR2}$/[2,1,low], {Software, DataMining}$_{BR2}$/[2,1,low]}

Note that $_{BR1,2}$/[10,2,low] in Example 3 is derived by counting RETs+DETs referenced by BR_1 and BR_2 in the corresponding Logical File (LF_1:{Computer, Software, HDUnit, MB, CPU, Videocard}).

4.5 Complexity of Transactional Functions

The following transactional functions address the user's capability to access configuration knowledge in ILFs and EIFs, i.e. maintaining, putting out and inquiring of configuration process-specific knowledge. In this context LFs are called FTRs (File Types Referenced - see Tables 3 and 4), i.e. a FTR denotes an ILF which is maintained or referenced by a transactional function or it denotes an EIF which is referenced by a transactional function.

---

[12] We use the notation [#RETs, #DETs, complexity], where the entries in Tables 1-5 are the result a corresponding data analysis (see Section 4.2).

[13] We assume that the number of component interface attributes is equal to the number of corresponding component attributes.

[14] We use the notation [#RETs+#DETs, #BRLF, complexity].

Example 4: Complexity of EI

EIs are represented by addTextedit(), deleteTextedit(), insertVideocard(), deleteVideocard(). For each of these operations only one FTR exists. For the purposes of our example we assume low complexity for each of those operations. ⌐

Example 5: Complexity of EO

EOs are represented by getPrice(), getMBPrice(). One FTR and one DET is referenced by the method getPrice(), i.e. the method has low complexity – the same holds for getMBPrice().⌐

Example 6: Complexity of EQ

EQs are exemplified by getHDUnitCapacity(HDUnit). Two FTRs and one DET are referenced, i.e. the method has low complexity. ⌐

	1-4 DET	5-15 DET	>16 DET
0-1 FTR	low	low	average
2 FTR	low	average	high
> 2 FTR	average	high	high

Table 3: Complexity of EIs

	1-5 DET	6-19 DET	>19 DET
0-1 FTR	low	low	average
2-3 FTR	low	average	high
> 3 FTR	average	high	high

Table 4: Complexity of EOs and EQs

4.6 Unadjusted Function Points

Based on the assignment of function points to different complexity classes (see Table 5) unadjusted function points (UFP) can be determined for the identified data- and transaction functions. These function points are company-specific and represent values calculated in phase 2 (Section 3, FP complexity classes).

Complexity	ILF	EIF	BR	EI	EO/EQ
low	0,25	0,15	0,25	0,25	0,25
average	0,5	0,5	1,5	1	1,5
high	1,5	1	2	1,5	2

Table 5: Determination of Function Points

Summing up these function points (see Table 6) results in 4,4 unadjusted FPs for our example which approximately corresponds to an effort of less than ½ MMs, i.e. 10 FPs are approximately equal to one MM. This first estimation (UFPs) must be adjusted using a set of adjustment factors (related to general system characteristics - GSC). GSCs are divided into two basic groups.

- Product characteristics, i.e. characteristics related to properties of the configuration application (e.g. requirements for distributed configuration support etc.).

- Project characteristics, i.e. characteristics related to management strategies and project team (e.g. how well are configuration concepts known by the team/customer?).

Adjusted Function Points (FPs) are determined as follows: FP=UFP*(sdev+(TDI*0.01)) represents the Total Degree of Influence calculated from GSCs. The relationship between FPs and MMs (10:1) is a constant in traditional software development, i.e. is used by well-known Function Point oriented effort estimation approaches [1]. In this formula *sdev* represents the standard deviation when analyzing the development time record sets (*sdev* in % of average project duration). *TDI* indicates additional efforts which can be expected in the current configurator project (e.g. implementation of

distributed configuration solutions or the implementation of Web-based configuration applications with special performance requirements[15]).

	low	average	high	sum
ILF	2	1	0	1
EIF	6	0	0	0,9
BR	3	0	0	0,75
EO/EQ	2/1	0	0	0,75
EI	4	0	0	1
UFPs				4,4

Table 6: UFPs for example application

5. EVALUATION

Applying the estimation concepts within a company requires the availability of data recording mechanisms integrated within the configurator development environment (the integration of such functionalities is shown in e.g. [13]). The presented concepts allow the provision of more concrete time and effort estimates for the customer. Experiences show that the application of the presented estimation concepts leads to a higher consciousness w.r.t. factors influencing implementation efforts and consequently to improved predictions of efforts. A very important influence factor w.r.t. the acceptance of an effort estimation method is its seamless integration into the given development environment. An automated data collection tool can provide such a functionality (such a feature has to be integrated by the corresponding configurator companies). Finally, FPA concepts are simple to use (a rather small set of concepts) – for this reason it can effectively be integrated into a company-specific development process.

6. RELATED WORK

The identification of sources of variations in effort estimation can significantly contribute to more reliable estimations for software projects [10]. In this paper configurator development is identified as such a source of variation which is tackled by adapting FPA to the special conditions of knowledge-based configurator development. There exists a number of approaches applying FPA to object-oriented software development (e.g. [8,14]). A direct application of these approaches to configurator development effort estimation results in significant and unacceptable deviations. The COSMIC [1] approach is a ISO standard effort estimation approach within the context of conventional software development projects. Although the method provides an interface for introducing additional measures, COSMIC does not explicitly take into consideration effort estimation support for knowledge-based systems development. Within the context of our projects we chose to apply conventional FPA, however the integration of our concepts into COSMIC is the subject of future work. The Feature Point approach (see [9]) is an extension to FPA which introduces (beside data functions and transactional functions) the complexity of algorithms as an additional parameter influencing effort estimation. Compared to our approach, Feature Points consider algorithms rather than business rules in knowledge bases. Effort estimation approaches in knowledge-based systems development provide a number of metrics (e.g. size metrics such as rule set density) but do not provide any experimental data to relate metrics to concrete effort sizes. [3] discuss different factors influencing development efforts in configurator development. A set of factors is presented influencing effort size in configuration projects – no relationship between those factors an concrete effort measures is defined.

7. CONCLUSIONS

We have shown the application and extension of Function Point Analysis (FPA) for effort estimation in the development of knowledge-based configuration systems. A framework for a company-specific application has been presented which reduces prediction error rates compared to the application of conventional FPA approaches. Using this approach, effort estimation techniques from conventional development can be integrated into development processes for knowledge-based (configuration) systems development. Further work will include the analysis of domain-specific complexity classes and the inclusion of discrete simulation models in order to further improve the accurateness of effort predictions.

---

[15] For reasons of space limitations we do not discuss the whole set of GSC.

8. REFERENCES

1. Abran, J-M. Desharnais, S. Oligny, St-Pierre D., and Symons C. (2003). COSMIC-FFP Measurement Manual. *The COSMIC Implementation Guide for ISO/IEC 19761:2003*.
2. Albrecht A. (1979). Measuring Application Development Productivity. In *IBM Applications Development Symposium*, Monterey, CA.
3. Aldanondo M. and Moynard G. (2002). Deployment of Configurator in Industry: Towards a Load Estimation. In *ECAI 2002 Workshop on Configuration*, pp. 125-130, Lyon, France.
4. Antoniol G., Calzolari F., Cristoforetti L., Fiutem R., and Caldiera G. (1998). Adapting Function Points to Object Oriented Information Systems. In Advanced Information Systems Engineering, CAISE'98, pp. 59-76, Monterey, CA.
5. Chandrasekaran B., Josephson J., and Benjamins R. (1999). What Are Ontologies, and Why do we Need Them? IEEE Intelligent Systems, 14(1):20-26.
6. Felfernig A., Friedrich G., and Jannach D. (2000). UML as domain specific language for the construction of knowledge-based configuration systems. International Journal on Software Engineering and Knowledge Engineering (IJSEKE), 10(4):449-469.
7. Felfernig A., Friedrich G., Jannach D., Stumptner M., and Zanker M. (2003). Configuration knowledge representations for Semantic Web applications. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM), 17:31-50.
8. Fetcke T., Abran A., and Nguyen T. (1998). Mapping the OO-Jacobson Approach into Function Point Analysis. In *Proceedings TOOLS '97*, Santa Barbara, CA.
9. Hastings T. (1995). Adapting Function Points to contemporary software systems - A review of proposals. Australian Software Metrics Association, editor, In *Proceedings 2nd Australian Conference on Software Metrics*.
10. Kemerer C.F. and Porter B.S. (1992). Improving the Reliability of Function Point Measurement: An Empirical Study. IEEE Transactions on Software Engineering, 18(11):1011-1024.
11. Rumbaugh J., Jacobson I., and Booch G. (1998). The Unified Modeling Language Reference Manual. Addison-Wesley.
12. Sabin D. and Weigel R. (1998). Product Configuration Frameworks - A Survey. In B. Faltings and E. Freuder, editors, IEEE Intelligent Systems, Special Issue on Configuration, volume 13,4, pages 50-58.
13. Sillitti A., Janes A., Succi G., and Vernazza T. (2003). Collecting, Integrating and Analyzing Software Metrics and Personal Software Process Data. In 29 EUROMICRO 2003 Conference "New Waves in System Architecture". IEEE.
14. Uemura T., Kusumoto S., and Inoue K. (2001). Function-point analysis using design specifications based on the Unified Modeling Language. Journal of Software Maintenance and Evolution: Research and Practice, 13:223-243.
15. Warmer J. and Kleppe A. (1999). The Object Constraint Language - Precise Modeling with UML. Addison Wesley Object Technology Series.
16. Wielinga B. and Schreiber G. (1997). Configuration Design Problem Solving. IEEE Expert/Intelligent Systems and their Applications, 12,2:49-56.

WORKLOAD ESTIMATION FORMULAE FOR
THE DEPLOYMENT OF COMMERCIAL CONFIGURATORS

Michel Aldanondo[1], Guillaume Moynard[2], Khaled Hadj-Hamou[1]

(1) Centre de Génie Industriel, Ecole des Mines d'Albi-Carmaux
Campus Jarlard, 81013 Albi CT Cedex 09, France
Michel.Aldanondo@enstimac.fr , Khaled.Hadj-Hamou@enstimac.fr
(2) Lapeyre-GME
2-4 rue André Karman, 93200 Aubervilliers, France
guillaume.moynard@lapeyre.fr

Abstract: This communication aims to estimate the workload, WL, required to deploy a configurator in industry. We consider: (i) a two steps approach including functional, f, (customer need) and physical, p, (bill-of-materials) configuration and (ii) a workload effort corresponding with configurable product modeling, m, and coding, c. The estimation formula has the following shape: $WL = WL_{m,f} + WL_{m,p} + WL_{c,f} + WL_{c,p}$ with $WL_{i,j} = F$ (factors).
First we explain the "Why ?" and "When ?" and introduce this problem. Then a second section presents a five steps estimation method that underlines the need to identify (i) configurable product factors and (ii) workload estimation formula. The third section proposes a set of factors characterizing configurable products and a shape for the workload estimation formula. Last section deals with the identification of the estimation formula. Twenty representative product cases are identified and submitted to configuration experts who provided workloads. Parametric identification techniques permits to derive a workload estimation formula.

Significance: Workload estimation is a key issue for the deployment of configuration software. This communication proposes an estimation method and relevant estimation formulae. As far as we know the proposed elements are the first in the configuration domain.

Keywords: configuration, configurator, configurator deployment, deployment workload estimation,

1. INTRODUCTION

This section addresses first the reasons of the work, then two main aspects of the problem are presented in order to introduce the estimation method, the last sub-section provides some scientific background.

1.1 Workload estimation for commercial configurator deployment: why and when.

As configuration software are more and more frequently used by companies providing customized products, almost all the ERP software propose now configuration modules integrated in their offer as reported in [1]. The main issue of this kind of configuration module is selling customized product, our propositions are therefore oriented to rather simple products and can not be considered for complex technical configuration.

The Lapeyre group, an European leader in industrial carpentry (windows, doors, kitchen furniture…) who proposed the

study some years ago, has been being a significant user of configuration for almost 10 years. For the Lapeyre group and

for many companies fighting in a mass-customization market, thinking about the deployment of a configuration

software is an important issue that frequently deals with the two following questions:

- for each configurator deployment, how many man-months of deployment capacity should I forecast ?
- what would be the resulting deployment cycle time ?
  It is clear that behind these two questions lies a "cost problem" or more accurately a "return on investment" problem.

Therefore this kind of questions are raised before the deployment of the configurator, when the company hesitates

between keeping some kind of "manual configuration" helped by spreadsheet and jumping to a rather fancy

configuration software that may be complex to set up and to maintain. This article does not deal with deployment cycle

time but focus on the first question concerning the deployment workload.

1.2 Main aspects of the deployment of Commercial Configurator

Most of commercial configurators operate in two steps relying on two distinct configurator modules.

The first one permits to collect the customer requirements and to check their consistency. This first step can be considered as a requirement configuration sub-process that relies on a functional or descriptive approach of the configurable product. As customers or sales representatives can not have a sufficient technical knowledge about the product (components, technical constraints between them…), the variables of the product generic model correspond mainly with product characteristics that are significant for them (power, height, color…). The model is therefore functional or descriptive and the configuration result is a configured product according to a functional or descriptive point of view.

The second one allows to derive from the result of the first step, possibly completed with some technical choices, the list of components or bill-of-materials of the product. This second step corresponds in fact with the "original" configuration problem introduced by [2] dealing with the product components and the constraints that can exist between them. Therefore the generic model is mainly physical and is linked with the first one in order to configure the bill-of-materials.

It is important to note that, according to the company needs, the deployment of these two modules is not necessary, some company situations can require only the first module and relevant deployment task. Therefore it is important to differentiate in the estimation approach what we call from now functional configuration (first module) and physical configuration (second module).

During a configurator deployment, the two main work-load consumer tasks are : (i) modeling the product and (ii) coding the model in the configurator.

The purpose of the modeling step is "to put on the paper" the configurable product. Normally, this activity should be

conducted without taking into account the configurator that will be deployed. But in fact, most of the configurator

vendors propose some modeling tools that fit their software. Some scientific modeling approaches can be found relying

on various frameworks can be found in [3] for constraint satisfaction problem approaches or [4] for oriented object

modeling.

The purpose of the coding step is "to enter the product model" in the configurator. In a perfect world, this activity

should be conducted by somebody that knows only about configuration and configurator utilization without any product

specific knowledge. As the product knowledge is gathered in the model on the "paper", the person in charge of

modeling should just translate the model into the coding entities existing in the configurator.

As these two tasks are frequently achieved by different company teams, we think that the workload estimation should consider separately these two tasks.

With respect to previous considerations, we aim to establish a formula that permit to estimate the deployment workload, WL, considering the two points of view: functional, _f_, (customer need) and physical, _p_, (bill-of-materials) configurations for the two tasks : configurable product modeling, _m_, and coding, _c_. The estimation formula would present therefore the following decomposition:

- $WL = WL_{m,f} + WL_{m,p} + WL_{c,f} + WL_{c,p}$

Where each $WL_{i,j}$ should have its own estimation formula. This permits to deduce the various kind of workload : modeling task WL_m, coding task WL_c, functional deployment WL_f and physical deployment WL_f as shown in figure 1.

	Functional	Physical	
Modeling	$WL_{m,f}$	$WL_{m,p}$	WL_m
Coding	$WL_{c,f}$	$WL_{c,p}$	WL_c
	WL_f	WL_p	WL

Figure 1 – Workload decomposition

Finally, an important point is relevant to the deployed configuration software. For this study, only one configuration software (provided by a configuration software company) is used, therefore the estimation function is valid for this single software. But, as each software has good and bad deployment capabilities, we assume that the proposed results can be considered as a first order of magnitude when used with other configurators.

1.3 Scientific background

If some scientific works have been achieved for this question for ERP deployment in terms of method and approaches, as far as we know, deployment workload estimation for configuration software is not very frequent in the open scientific literature.

Many configurator vendors have got their own deployment method and approaches, but very few of them have been presented or discussed, some elements can be found in [5] but they do not deal with deployment workload estimation.

Some ideas about our problem can be found in [6] and this communication follows the reported elements with a first set of quantified results.

Few estimation works in the field of product design are closer to our estimation goal, [7] and [8] synthesizes various estimation formulae for product development. The shapes of the estimation formulae that will be proposed will be extrapolate from these works.

2. ELEMENTS FOR AN ESTIMATION METHOD

As explained in the first sub-section, the estimation need appears before the configurator deployment and most of the time before the decision of buying the configurator software. Consequently the estimation task itself must be quickly conducted (order of magnitude less than a week) through various meetings gathering actors of concerned company teams (at least marketing, selling, design, manufacturing). Our estimation formulae must be understand as an element of the following five steps estimation method.

- Configurable products identification

The first step goal is to decide about the configurable product set. This point is very delicate because it corresponds with the identification of an appropriate level of genericity of the product. A high level of genericity reduce the number of configurable products but increases its complexity while a low genericity provides the opposite result. Let us consider the following example : (1) the product can be window or door, (2) it can be used indoor or outdoor, (3) it can be made of wood, PVC or fiberboard, and (4) it can have three styles : modern, classic or rustic. The problem is to decide about the number of configurable products, between one (able to represents all the diversity) and 36 (one for each combination). We will not go deeper in this problem, but indicates that some ideas are reported in [5].

- Identification of a product representative sample

Once the set of configurable product is identified, it is necessary to identify one among them that is representative of the whole set. By representative, we mean a product that can be compared with all other products during the fifth step of extrapolation. As different company teams collaborate for this identification, this step is not obvious. Frequently, the representative sample for marketing and selling teams is different from the one proposed by design and manufacturing and some compromise must be found.

- Quantification of factors values for the configurable product sample

The product sample is then characterized by factors. We will see later that these factors aim to represent some "size" and "complexity" of the configurable product. As deployment is also strongly dependant of the company level of organization, it is also necessary to characterize the company where the deployment should take place.

- Work-load estimation of the deployment of the configurable product sample

The factors values are then inputted in the estimation formula and provide the workload value for the deployment of the configurable product sample, WL_{sample}.

- Extrapolation of the workload to the complete set of configurable products

The deployment workload of the sample is then extrapolated for each configurable product. This is done thanks to an extrapolation factor quantified for each product with respect to the sample, $Fex(WL_{prod-x}, WL_{sample})$. Learning effect and reusability could be also taken into account. Most of the time, learning progressively reduces workload when the number of deployed products increases. Reusability permits to take into account that some parts of model and/or some parts of code established for a given product can be sometimes reused in other products.

This way to operate, and more accurately steps 3 and 4, underlines the needs of (i) a list of factors able to characterize the product and (ii) an estimation formulae able to quantify the deployment workload. Next section is concerned with the definition of the factors and the shape of the estimation formulae.

3. ESTIMATION FACTORS AND FORMULAE

3.1 Workload deployment factors

Three kinds of factors have been identified. The first kind is independent of the product and characterizes the deployment environment. The second kind characterizes the configurable product from a functional point of view while the third one is concerned by the physical point of view. As we target a first order of magnitude estimation that should be achieved rapidly, it is a key point to identify a little number of factors at least less than ten. Consequently, the accuracy of the load estimation is expected to be around +/- 50%.

1. Deployment environment factor

All aspects relevant to the deployment environment are aggregated in one single factor, CP. This factor characterizes the level of organization of the company and the adequacy of actors of the company teams involved in the project. This factor is quantified by the person in charge of the estimation.

Company organization level characterizes the general behavior of the company in front of any software deployment. This aspect can be taken into account when estimating the workload deployment of ERP or PDM systems for example and aims to take into account at least : management capability, sub-contracting level and the fact that the company is multi-lingual and/or multi-located. Of course, a company with high management capability, no subcontracting, a single language and a single location will present the best situation for software deployment.

Actors adequacy characterizes the general organization and knowledge of the persons that could be involved in the configurator deployment. They are more specific to configurator deployment that the previous factors. The following elements should be considered :
- product knowledge : it characterizes if a single person with a good global knowledge level can be easily identified for each of major knowledge field (marketing, selling, design and manufacturing),
- modeling knowledge : it takes into account if a person in the company has already made some generic modeling, or if some models (whatever the formalism is) already exist for functional and physical configuration,
- Configuration model coding know-how : it characterizes if a person in the company has already implemented a product configuration model in a configurator, a spreadsheet or in any computer language.

In order to avoid a detailed quantification of all these aspects and an aggregation that is not at all obvious, the quantification of the factor CP is done with some kind of classification that aggregates the previous aspects in the following way :

- if the deployment requires a workload WL in a company with an "average" deployment environment factor (CP=1),
- we consider
  - that 10% of the best organized companies would need less than the half of WL (CP = 0.5),
  - that 10% of the worst organized companies would need more than the double of WL (CP=2).

This roughly means that the deployment environment factor can modulate an average workload either by dividing it or multiplying it by two. Interpolation is of course possible and the impact on this factor will be modulated by a parameter in the estimation formulae computing $WL_{m,f}$, $WL_{m,p}$, $WL_{c,f}$ and $WL_{c,p}$.

2. Factors relevant to the product from a functional point of view

As explained in the first section, functional configuration relies on a product generic model gathering products characteristics. Constraints, representing allowed/excluded combinations of characteristics values, limit the space of possible configurations. The three following factors have been identified and must be quantified for the representative product sample.

The first two factors that characterizes a kind of "size" of the generic model are :

- NbVar : represents the number of characteristics on which a customer requirement can be expressed,
- NbVal : represents the average number of possible values for the characteristics.

The third factor tries to characterize some kind of complexity of the product. If all characteristics are independent, there

is no constraint and all options are additive and therefore the product is considered simple. If many constraints are

present, options are not anymore additive and the product is considered complex. We therefore propose a ratio between

the number constraints and the number of characteristics aiming to represent some kind of constraint density. It is

important to note that we consider that one constraint gathers all the allowed combination of values between a given set

of characteristics and not a single combination of values.

- DstCst : represents the number of constraints divided by the number of characteristics.

These three factors should be taken into account in each formula relevant to functional configuration computing $WL_{m,f}$, and $WL_{c,f}$.

3. Factors relevant to the product from a physical point of view

As explained in the first section, physical configuration relies on a product generic model corresponding with a kind of generic hierarchical bill-of-materials. We consider that a generic bill-of-materials is a set of items. Items correspond at the upper levels with generic sub-assemblies (that have a generic bill-of-materials themselves) and at the lowest level with standard components (that are standard with frozen characteristics). Therefore :

- an upper level generic sub-assembly can gather a set of lower level generic sub-assemblies and/or a set of lowest level standard components,
- a generic link identifies each couple of items (upper level generic sub-assembly, lower level generic sub-assembly) or (upper level generic sub-assembly, lowest level standard component).

A generic link permits to modulate the existence of the child item. This generic link is in fact a constraint that depends on the products characteristics of the functional model.

For this estimation work we consider that constraints between items in the physical generic model are in fact taken into account in the functional generic model. This means that when a product is functionally configured, only one bill-of-materials can be identified. Therefore pure technical choices, that do not interest the customer, dealing with product composition are not taken into account.

Consequently, the product physical point of view factors characterizes only the "size" aspect of the product without

taking into any kind of complexity. We consider that the size aspect is only relevant to the number of generic links that

exist in the generic bill-of-materials. Two factors are nevertheless identified in order to differentiate lowest level generic links from upper level ones :

- NbLup : represents the number of generic links between upper level generic sub-assembly, lower level generic sub-assembly,
- NbLlo : represents the number of generic links between upper level generic sub-assembly, lowest level standard component.

These two factors should be taken into account in each formula relevant to physical configuration computing $WL_{m,p}$, and $WL_{c,p}$.

4. Conclusion

Six factors are identified :
- One common to the four estimation formula : CP,
- Three dedicated to the functional point of view: NbVar, NbVal, DstCst,
- Two dedicated to the physical point of view: NbLup, NbLlo.

Next section proposes shape for estimation formulae that operate with these factors.

3.2 Estimation formula shapes

The proposed shapes for the estimation formulae are extrapolated from the works of [7] and [8]. The first work dealing with product development considers a multiplication of four factors : product size, product complexity, company productivity and reuse possibilities

Size and complexity are taken into account in the factors relevant to the functional and physical points of view of the product. We consider that productivity can be associated with the deployment environment factor and that reuse possibilities are taken into account when extrapolating in the fifth step of the estimation method.

We therefore propose to keep this idea and add to each factor a power parameter (αi for functional and βi for physical points of view) that permits to modulate the effect of the factor and an overall gain parameter. This provides :

- for the functional deployment : $WL_f = NbVar^{\alpha 1} * NbVal^{\alpha 2} * DstCst^{\alpha 3} * CP^{\alpha 4} * \alpha 5.$
- for the physical deployment : $WL_p = NbLlo^{\beta 1} * NbLup^{\beta 2} * CP^{\beta 3} * \beta 4.$

For functional deployment formula, when there is no constraint (all options or variants are additive) the factor DstCst equals zero and the resulting workload also. Therefore as proposed in the work of [8] in order to highlight complexity the complexity factor is modified in an exponential factor. This provide the following formula :

- for the functional deployment : $WL_f = NbVar^{\alpha 1} * NbVal^{\alpha 2} * \alpha 3^{DstCst} * CP^{\alpha 4} * \alpha 5.$

We therefore propose one estimation formula for each point of view for each configuration module deployment (functional and physical). But each of these two formulae must be identified for the two deployment task modeling and coding. Formulae identification will therefore provide four estimation formulae as shown in figure 2.

	Functional	Physical
Modeling	$WL_{m,f} =$ $NbVar^{\alpha 1,m} * NbVal^{\alpha 2,m} * \alpha 3, m^{DstCst} * CP^{\alpha 4,m} * \alpha 5, m$	$WL_{m,p} =$ $NbLlo^{\beta 1,m} * NbLup^{\beta 2,m} * CP^{\beta 3,m} * \beta 4, m$
Coding	$WL_{c,f} =$ $NbVar^{\alpha 1,c} * NbVal^{\alpha 2,c} * \alpha 3, c^{DstCst} * CP^{\alpha 4,c} * \alpha 5, c$	$WL_{c,p} =$ $NbLlo^{\beta 1,c} * NbLup^{\beta 2,c} * CP^{\beta 3,c} * \beta 4, c$

Figure 2 – Workload estimation formulae

126

4. WORKLOAD ESTIMATION FORMULAE IDENTIFICATION

This section is dedicated to the identification of the four estimation formulae. Data collection is first addressed followed by formulae identification, a discussion terminates the section.

4.1 Data collection

In order to collect data, a set of cases is identified. A case is a set of factors with a defined value for each. These cases are proposed to different configuration experts of the Lapeyre Company who provide values for the four workloads.

1. Cases definition.

First a set of possible values has been defined for the six factors. For each factor three values haves been selected : an average one, an upper and a lower one. Upper and lower bounds were defined for each factor, in a way that around 80% of Lapeyre configurable products could be located between upper and lower bounds. This provide the following factors values.

	Lower	Average	Upper
CP	0.5	1	2
NbVar	10	20	40
NbVal	5	10	20
DstCst	0.5	1	2
NbLup	25	50	100
NbLlo	250	500	1000

Once these values are identified, cases are defined separately for the functional and physical configuration. For each point of view :
- three cases present the same level for each factor (cases 1 to 3),
- for each factor, two cases with upper and lower levels while all the others remain at the average level (remaining cases).

This gives the following table of cases.

Functional configuration						Physical configuration			
Case	CP	NbVar	NbVal	DstCst		Case	CP	NbLup	NbLlo
1	1	20	10	1		1	1	50	500
2	2	40	20	2		2	2	100	1000
3	0.5	10	5	0.5		3	0.5	25	250
4	2	20	10	1		4	2	50	500
5	1	40	10	1		5	1	50	1000
6	1	20	20	1		6	1	100	500
7	1	20	10	2		7	0.5	50	500
8	0.5	20	10	1		8	1	50	250
9	1	10	10	1		9	1	25	500
10	1	20	5	1					
11	1	20	10	0.5					

2. Data collection

These cases were submitted to experts. In the Lapeyre group, actors in charge of modeling are most of the time responsible of coding their models, but most of the experts are specialized in either product functional knowledge or physical knowledge. This comes mainly form the fact that functional modeling is achieved by marketing and sale teams while physical modeling is done by actors from design and manufacturing teams. The involvement of the experts in configuration varies between at least one year and more than three years. Cases were submitted to eight experts for functional configuration and six experts for the physical one. For each case, the mean of the estimated workloads is shown bellow, the unit for workload is work-day.

Functional configuration				Physical configuration		
Case	$WL_{m,f}$	$WL_{c,f}$		Case	$WL_{m,p}$	$WL_{c,p}$
1	2.13	4.25		1	4.33	5.08
2	5.94	12.25		2	10.25	9.5
3	1.2	2.06		3	1.83	2.33
4	4.07	7.06		4	7.5	7.17
5	3	7.63		5	5.67	5.83
6	2.56	5.63		6	6.33	7.33
7	3.31	6.06		7	3.00	3.83
8	1.31	3.81		8	3.33	3.33
9	1.38	2.94		9	3.33	3.17
10	1.69	3.56				
11	1.69	3.38				

4.2 Formulae identification

Formulae identification was done thanks to mean square regression with the mathematical package Scilab (Trademark of INRIA). Each set of data relevant to each workload and the corresponding formula shape were inputted and the formulae of figure 3 were identified.

	Functional	Physical
Modeling	$WL_{m,f} = NbVar^{0.29} * NbVal^{0.06} * 1.31^{DstCst} * CP^{0.72} * 0.62$	$WL_{m,p} = NbLlo^{0.24} * NbLup^{0.36} * CP^{0.61} * 0.25$
Coding	$WL_{c,f} = NbVar^{0.6} * NbVal^{0.17} * 1.18^{DstCst} * CP^{0.4} * 0.41$	$WL_{c,p} = NbLlo^{0.17} * NbLup^{0.46} * CP^{0.38} * 0.27$

Figure 3 – Workload estimation formulae

Comparison between expert estimation and formulae calculation are shown bellow.

Functional configuration						Physical configuration				
	Expert source		Calculated				Expert source		Calculated	
Case	$WL_{m,f}$	$WL_{c,f}$	$WL_{m,f}$	$WL_{c,f}$		Case	$WL_{m,p}$	$WL_{c,p}$	$WL_{m,f}$	$WL_{c,f}$
1	2.13	4.25	2.25	4.28		1	4.33	5.08	4.56	4.92
2	5.94	12.25	6.14	11.33		2	10.25	9.5	10.61	10.03
3	1.2	2.06	0.94	1.76		3	1.83	2.33	1.96	2.41
4	4.07	7.06	3.67	5.65		4	7.5	7.17	6.99	6.34
5	3	7.63	2.75	6.47		5	5.67	5.83	5.39	5.64
6	2.56	5.63	2.34	4.82		6	6.33	7.33	5.86	6.78
7	3.31	6.06	2.93	5.06		7	3.00	3.83	2.98	3.81
8	1.31	3.81	1.37	3.25		8	3.33	3.33	3.86	4.28
9	1.38	2.94	1.83	2.84		9	3.33	3.17	3.65	3.56
10	1.69	3.56	2.15	3.81						
11	1.69	3.38	1.96	3.94						

The average percentage of the absolute value of the error between the expert source and the calculated ones are around 12% and 13% for the functional configuration and 7% and 8% for the physical one.

4.3 Discussion

From the presented results it is possible to make some remarks.

Considering the parameters related to the deployment environment factor CP, it is interesting to note that its values are lower for the coding task ($\alpha4,c = 0.4$ and $\beta3,c = 0.38$) than for the modeling task ($\alpha4,m = 0.72$ and $\beta3,m = 0.61$). This factor influences more modeling than coding. The fact that modeling requires much more discussions and compromises than coding can explain this conclusion.

A same kind of conclusion can be drawn about the complexity factor relevant to functional configuration. Its parameter value is lower for coding ($\alpha3,c = 1.18$) than for modeling ($\alpha3,m = 1.31$). When the complexity of the product increases, it sounds logical that the effort related to the design of the model increases much more that the one concerning model coding.

In terms of "size" of the product from a functional point of view, it can be noticed that the number of characteristics NbVar has a more significant influence ($\alpha1,m = 0.29$ and $\alpha1,c = 0.6$) than the one characterizing the possible values NbVal ($\alpha2,m = 0.06$ and $\alpha2,c = 0.17$). Here also, it sounds rather logical that the number of characteristics influences more modeling and coding task size than the number of possible choices.

From parameters relevant to the number of links for physical configuration, it can be noted that the parameters relevant to the links between generic sub-assemblies ($\beta2,m = 0,36$ and $\beta2,c = 0,46$) are higher than the ones relevant to the lowest level links ($\beta1,m = 0,24$ and $\beta1,c = 0,17$). In fact when factor values are taken into account, impacts on modeling workloads are almost the same while impact on coding workload is much more important for links between generic sub-assemblies. This difference is probably due to code testing requirements that are much stronger for high level generic links.

The proposed estimation functions must in any case considered as an "order of magnitude" workload estimation. In order to try to verify the order of magnitude accuracy, these formulae have been applied to the entire set of configurable products of the Lapeyre group.

Four hundreds functional models and twenty physical models have been designed and coded and are currently in used. A representative sample permits to establish that 8 and 18 work-days were necessary for deploying respectively the functional and the physical of the product sample. Extrapolation, without consideration reuse, lead to a total workload around 15 man-years. Comparing this result with the number of people involved in configuration and the time spent, this workload sounds fine for an order of magnitude. According to the person at the head of the activity, it is sure that the work force involved is greater than 10 man-years but in any case lower than 25.

5. CONCLUSION

The goal of this communication was to propose elements for estimating the workload required for the deployment of a commercial configurator. An estimation method relying on a set of six factors and four estimation formulae has been proposed. These reported elements must be considered as a very first set of results able to estimate an order of magnitude for the deployment of commercial configurators. Different ways to improve accuracy can be investigated.

A way to increase the accuracy of the estimation formulae is to replace expert estimation by workload measurements on real deployment projects. This was the initial idea when the work started. But the delicate point is to gather reliable data relevant to terminated project. This is, nevertheless, the only way to avoid the lack of objectivity of experts.

An improvement of the estimation method could be to replace the workload estimation of the product sample by measurements achieved on real modeling and coding of the sample. Only workload extrapolation would be therefore necessary and would require some work.

Workload estimation for configurator deployment is not an easy task. The proposed elements helps a little bit the Lapeyre Group when facing configuration deployment for a new product line or for a new facility. They permit some kinds of validation of the intuitive workload provided by the persons who have the largest experience in configuration deployment.

6. REFERENCES

1. Gartner Group. Supercharge the Sales Effort: Sales Application for Large Enterprise. <u>Strategic Report Analysis n° R-14-4280</u>, Gartner Research, September 2001.

2. Mittal, S. and Frayman, F. (1989). Towards a generic model of configuration tasks. <u>Proceedings of the International Joint Conference on Artificial Intelligence 89</u> pp 1395-1401, Morgan Kaufman Publishers, Detroit, USA.

3. Aldanondo, M., Hadj-Hamou, K., Moynard G., and Lamothe, J., (2003). Mass customization and configuration: Requirement analysis and constraint based modeling propositions. <u>Integrated Computer-Aided Engineering</u>. IOS Press Vol. 10, n° 2, pp 177-189.

4. Felfernig, A., Friedrich G., Jannach D., (2000). UML as domain specific language for the construction of knowledge-based configuration systems. <u>International Journal of Software Engineering and knowledge Engineering</u>, Vol. 10**,** n° 4, pp 449-469.

5. Mortensen, N.H., Yu, B., Skovgaard, H.S. and Harlou, U. (2000). Conceputal modeling of product families in configuration projects. <u>ECAI-2000 Workshop on Configuration</u>, pp. 68-73, Berlin, Germany.

6. M. Aldanondo, M. and Moynard G. (2002). Deployment of Configurator in Industry : Towards a Load Estimation. <u>ECAI-2002 Workshop on Configuration</u>, pp 125-130, Lyon, France.

7. Jacome, M. F. and Lapinskii, V. (1997). NREC: Risk assessment and planning of complex designs. <u>IEEE Design & Test of Computers</u>, Vol. 14, n°. 1, pp 42-49.

8. Bashir, H.A. and Thomson, V. (1999). Metrics for Design Projects: A Review. <u>Design Studies</u>, Vol. 20, n° 3, pp 263-277.

FAST BACKTRACK-FREE PRODUCT CONFIGURATION USING A PRECOMPILED SOLUTION SPACE REPRESENTATION

Tarik Hadzic, Sathiamoorthy Subbarayan, Rune M. Jensen, Henrik R. Andersen, Jesper Møller[*], and Henrik Hulgaard[*]

Department of Innovation	Configit Software A/S[*]
IT University of Copenhagen	Vermundsgade 38 B
Glentevej 67	DK-2100 Copenhagen Ø
2400 Copenhagen NV	Denmark
Denmark	+45 7022 6700
+45 3816 8888	Email: {jm, henrik}@configit-software.com
Email: {tarik, sathi, rmj, hra}@itu.dk	

In this paper we describe a two-phase approach to interactive product configuration. In the first phase, a compressed symbolic representation of the set of valid configurations (the *solution space*) is compiled offline. In the second phase, this representation is embedded in an online configurator and utilized for fast, complete, and backtrack-free interactive product configuration. The main advantage of our approach compared to online search-based approaches is that we avoid searching for valid solutions in each iteration of the interactive configuration process. The computationally hard part of the problem is fully solved in the offline phase given that the produced symbolic representation is small. The employed symbolic representation is Binary Decision Diagrams (BDDs). More than a decade of research in formal verification has shown that BDDs often compactly encode formal models of systems encountered in practice. To our experience this is also the case for product models. Often the compiled BDD is small enough to be embedded directly in hardware. Our research has led to the establishment of a spin-off company called Configit Software A/S. Configit has developed software for writing product models in a strongly typed language and has patented a particularly efficient symbolic representation called Virtual Tables.

1. INTRODUCTION

The focus in manufacturing industry has shifted from mass production to mass customization. Companies continually have to offer more product variants with greater flexibility. At the same time, the rapid development of information technology has significantly increased the complexity of each individual product. These changes have led to a situation where the industry is losing track of the functionality of their products. It has become common practice to ship mobile phones and software that is close to impossible to setup correctly even for expert customers.

Efficient tools are needed to handle the increasing complexity of products. Product configurators are one such class of tools. Given a set of rules defining the set of valid configurations (the *solution space*) of the product, the configurators guide sales people and users to find a valid and desirable configuration of the product. Most configurators are based on searching online in the solution space [8, 9, 14]. This may work well for many products, but it is impossible to guarantee that the search time is polynomially bounded with the size of the product model, since finding just a single valid configuration is NP-complete. This means that search-based configurators sometimes may have undesirable long response times. Moreover, the performance of these tools often depends on how the rules are written, and it can be very difficult to write rules that work well in practice.

In this paper we describe an alternative approach to product configuration based on a precompiled representation of the solution space. The approach has two phases. The first phase is offline and consists of compiling the product rules into a Binary Decision Diagram (BDD) [2] representing the solution space. BDDs are a canonical representation of Boolean functions. During the last 15 years, they have been applied successfully in formal verification and other areas of computer science to represent formal models of very large systems [3]. Our experience is that BDDs also compactly encode the solution space of industrial products. Since the compiled BDD is canonical, it only depends on what Boolean function the

product rules represent and not on how the rules are written. This gives the rule writer freedom to choose a format of the rules that naturally represents the behavior of the product.

In the second phase, the compiled BDD is used online in an interactive configurator. In each iteration of the interactive configuration process, specialized BDD algorithms compute the set of possible ways the current partial configuration can be extended to a valid product. The interactive configuration process is complete and backtrack-free. The user can choose freely between any valid configuration and is prevented from reaching dead-ends of impossible configurations. More importantly, the worst-case response time only grows polynomially with the size of the BDD. Thus, the computationally hard part of the configuration problem is fully solved in the offline phase given that the compiled BDD is small. Surprisingly this is often the case even for complex products with long compilation times.

Our research has led to the establishment of a spin-off company called Configit Software A/S. Configit has improved the BDD-based technique and patented a particularly efficient symbolic representation called Virtual Tables (VTs). A VT is an XML file that in addition to the symbolic representation holds the definition of the information it stores. VTs can be embedded in a wide variety of products ranging from web-configurators to electronic products.

The remainder of the paper is organized as follows. In Section 2, we formally define product configuration and describe the interactive configuration process. In Section 3, we show how to encode a solution space symbolically as a Boolean function and illustrate how this can be done with BDDs. In Section 4, we briefly introduce Configit's approach and special features. Section 5 presents experimental work. Related work is discussed in Section 6. Finally in Section 7, we conclude and consider directions for future work.

2. INTERACTIVE PRODUCT CONFIGURATION

We can think of product configuration as a process of specifying a product defined by a set of attributes, where attribute values can be combined only in predefined ways. Our formal definition captures this as a mathematical object with three elements: variables, domains for the variables defining the combinatorial space of possible assignments and formulae defining which combinations are valid assignments. Each variable represents a product attribute, variable domain refers to the options available for its attribute and formulae specify the rules that the product must satisfy.

Definition 1. A *configuration problem* C is a triple (X, D, F), where X is a set of variables x_1, x_2, K, x_n, D is a Cartesian product of their finite domains $D_1 \times D_2 \times K \times D_n$ and $F = \{f_1, f_2, K, f_m\}$ is a set of propositional formulas over atomic propositions $x_i = v$, where $v \in D_i$, specifying conditions that the variable assignments have to satisfy.

Each formula f_i is a propositional expression φ inductively defined by

$$\varphi \equiv x_i = v \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \neg \varphi,$$

where $v \in D_i$. We will use the abbreviation $\varphi \Rightarrow \psi \equiv \neg \varphi \vee \psi$ for logical implication. For a configuration problem C, we define the *solution space* $S(C)$ as the set of all *valid configurations*, i.e. the set of all assignments to the variables X that satisfy the rules F. Many interesting questions about configuration problems are hard to answer. Just determining whether the solution space is empty is NP-complete, since we can reduce the Boolean satisfiability problem to it in polynomial time [10].

As an example consider specifying a T-shirt by choosing the color (black, white, red or blue), the size (small, medium or large) and the print ("Men In Black" - MIB or "Save The Whales" – STW). There are two rules that we have to observe: if we choose the MIB print then the color black has to be chosen as well, and if we choose the small size then the STW print (including a big picture of a whale) cannot be selected as the large whale does not fit on the small shirt.

The configuration problem (X, D, F) of the T-shirt example consists of variables $X = \{x_1, x_2, x_3\}$ representing color, size and print. Variable domains are $D_1 = \{black, white, red, blue\}$, $D_2 = \{small, medium, large\}$ and $D_3 = \{MIB, STW\}$. The two rules translate to $F = \{f_1, f_2\}$ where f_1 is $(x_3 = MIB) \Rightarrow (x_1 = black)$ and f_2 is $(x_3 = STW) \Rightarrow (x_2 \neq small)$. There are $|D_1||D_2||D_3| = 24$ possible assignments. Eleven of these assignments are valid configurations and they form the solution space shown in Figure 1.

(*black, small, MIB*)	(*black, large, STW*)	(*red, large, STW*)
(*black, medium, MIB*)	(*white, medium, STW*)	(*blue, medium, STW*)
(*black, medium, STW*)	(*white, large, STW*)	(*blue, large, STW*)
(*black, large, MIB*)	(*red, medium, STW*)	

Figure 1. Solution space for the T-shirt example.

When we talk about *interactive configuration,* we are referring to the process of a user interactively tailoring a product to his specific needs by using supporting software called a *configurator*. Every time the user assigns a value to a variable, the configurator restricts the solution space by removing *all* assignments that violate this new condition, reducing the available

user choices to only those values that appear in *at least* one configuration in the restricted solution space. The user keeps selecting variable values until only one configuration is left. The algorithm in Figure 2 illustrates this interactive process.

INTERACTIVE-CONFIGURATION(C)

1   $Sol \leftarrow S(C)$

2   **while** $|Sol| > 1$

3   **do choose** $(x_i = v) \in$ VALID-ASS(Sol)

4         $Sol \leftarrow Sol \text{ I } D_{x_i = v}$

Figure 2. Interactive configuration procedure.

The VALID-ASS(Sol) procedure in line 3 extracts the set of valid assignments (choices) from the solution space *Sol*. We restrict the solution space in line 4 by intersection with $D_{x_i = v} = D_1 \times \text{K } D_{i-1} \times \{v\} \times D_{i+1} \times \text{K } D_n$, which effectively enforces that only those tuples with value v for x_i remain in the solution space.

This behavior of the configurator enforces a very important property of interactive configuration called *completeness of inference*. The user cannot pick a value that is not a part of a valid solution, and furthermore, a user is able to pick *all* values that are part of at least one valid solution. These two properties are often not satisfied in existing configurators, either exposing the user to backtracking or making some valid choices unavailable.

In the T-shirt example, the assignment $x_2 = small$ will, by the second rule, imply $x_3 \neq STW$ and since there is only one possibility left for variable x_3, it follows that $x_3 = MIB$. The first rule then implies $x_1 = black$. Unexpectedly, we have completely specified a T-shirt by just one assignment. Actually, the configurator just deletes all configurations that do not satisfy $x_2 = small$ and discovers that a solution space is reduced to just one tuple: $(black, small, MIB)$.

From the user's point of view, the configurator responds to the assignment by calculating valid choices for undecided variables. It is important that the response time is very short, offering the user truly interactive experience. The demand for short response-time and completeness of inference is difficult to satisfy due to the hardness of the configuration problem.

3. TWO PHASE APPROACH

Since checking whether the solution space is empty is NP-complete, it is unlikely that we can construct a configurator that takes a configuration problem C and guarantees a response time that is polynomially bounded with respect to the size of C.

Our approach is offline to compile the solution space of a configuration problem to a representation that supports fast interaction algorithms. The idea is to remove the hard part of the problem in the offline phase. This will happen if the compiled representation is small. We cannot always avoid exponentially large representations. However, for most real-world problem instances, we get small representations and therefore fast interaction algorithms. Furthermore, after the compilation is finished, we know the size of the solution space representation. Therefore we are able to precisely predict the running time of the interaction algorithms.

3.1 Symbolic Solution Space Representation

Our configuration problem C can be efficiently encoded using Boolean variables and Boolean functions. We assume that domains D_i contain successive integers starting from 0. For example, we encode $D_2 = \{small, medium, large\}$ as $D_2 = \{0,1,2\}$. Let $l_i = \lceil \lg |D_i| \rceil$ denote the number of bits required to encode a value in domain D_i. Every value $v \in D_i$ can be represented in a binary format and therefore seen as a vector of Boolean values $\vec{v} = (v_{l_i-1}, \text{K }, v_1, v_0) \in \mathbf{B}^{l_i}$. Analogously, every variable x_i can be encoded by a vector of Boolean variables $\vec{b} = (b_{l_i-1}, \text{K }, b_1, b_0)$. Now, the formula $x_i = v$ can be represented as a Boolean function given by the expression $\vec{b} = \vec{v}$ i.e. $b_{l_i-1} = v_{l_i-1} \wedge \text{K } \wedge b_1 = v_1 \wedge b_0 = v_0$.

In the T-shirt example, $D_2 = \{small, medium, large\}$ and $l_2 = \lceil \lg 3 \rceil = 2$, so we can encode $small \in D_2$ as 00 $(b_1 = 0, b_0 = 0)$, *medium* as 01 $(b_1 = 0, b_0 = 1)$ and *large* as 10 $(b_1 = 1, b_0 = 0)$. This translation to a Boolean domain is not surjective, i.e. not every combination of assignments to Boolean variables $b_{l_i-1}, \text{K }, b_1, b_0$ yields a valid value $v \in D_i$. For example, the combination 11 does not encode a valid value in D_2. Therefore we introduce a Boolean constraint (a so called *domain constraint)* that forbids these unwanted combinations $F_D = \wedge_{i=1}^{n}(\vee_{v \in D_i} x_i = v)$. Furthermore, we define a translation function τ that maps a propositional expression φ to the Boolean function it represents

$$\tau(\varphi) : \prod_{i=1}^{n} \mathbf{B}^{l_i} \to \mathbf{B} .$$

The translation is defined inductively as follows

$$\tau\left(x_i = v\right) \equiv \left(b_i^p = v^p\right)$$
$$\tau\left(\varphi \wedge \psi\right) \equiv \tau\left(\varphi\right) \wedge \tau\left(\psi\right)$$
$$\tau\left(\varphi \vee \psi\right) \equiv \tau\left(\varphi\right) \vee \tau\left(\psi\right)$$
$$\tau\left(\neg\varphi\right) \equiv \neg\tau\left(\varphi\right).$$

Finally, we are able to express a Boolean function representation $S'(C)$ of the solution space $S(C)$

$$S'(C) \equiv \wedge_{i=1}^m \tau\left(f_i\right) \wedge \tau\left(F_D\right).$$

The interactive process of product configuration can be represented using the already described procedure (Figure 2), but now using the Boolean representation of the solution space. The resulting algorithm is shown in Figure 3.

INTERACTIVE-CONFIGURATION(C)
1   $Sol \leftarrow S'(C)$

2   **while** $\left|Sol\right| > 1$

3   **do choose** $\left(x_i = v\right) \in$ VALID-ASS(Sol)

4         $Sol \leftarrow Sol \wedge \tau\left(x_i = v\right)$

Figure 3. Boolean version of interactive configuration.

3.2 Binary Decision Diagrams

A reduced ordered Binary Decision Diagram (BDD) is a rooted directed acyclic graph representing a Boolean function on a set of linearly ordered Boolean variables. It has one or two terminal nodes labeled 1 or 0 and a set of variable nodes. Each variable node is associated with a Boolean variable and has two outgoing edges *low* and *high*. Given an assignment of the variables, the value of the Boolean function is determined by a path starting at the root node and recursively following the high edge, if the associated variable is true, and the low edge, if the associated variable is false. The function value is *true*, if the label of the reached terminal node is 1; otherwise it is *false*. The graph is ordered such that all paths respect the ordering of the variables. A BDD representing the function $f(x_1, x_2) = x_1 \vee \neg x_1 \wedge \neg x_2$ is shown Figure 4a.

A BDD is reduced such that no two distinct nodes u and v are associated with the same variable and low and high successors (Figure 4b), and no variable node u has identical low and high successors (Figure 4c).

(a)   (b)   (c)
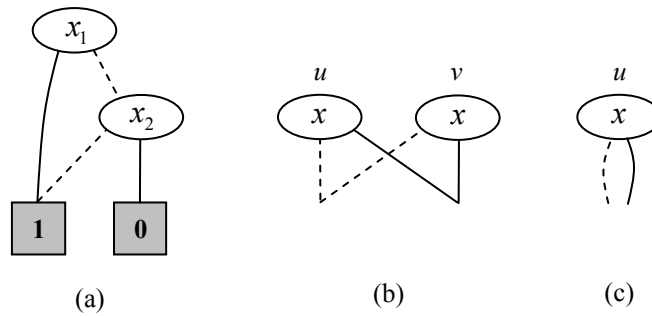
Figure 4. (a) A BDD representing the function $f(x_1, x_2) = x_1 \vee \neg x_1 \wedge \neg x_2$. High and low edges are drawn with solid and dashed lines, respectively. (b) Nodes associated with the same variable with equal low and high successors will be converted to a single node. (c) Nodes causing redundant tests on a variable are eliminated.

Due to these reductions, the number of nodes in a BDD for many functions encountered in practice is often much smaller than the number of truth assignments of the function. Another advantage is that the reductions make BDDs canonical [2]. Large space savings can be obtained by representing a collection of BDDs in a single multi-rooted graph where the sub-graphs of the BDDs are shared. Due to the canonicity, two BDDs are identical if and only if they have the same root. Consequently, when using this representation, equivalence checking between two BDDs can be done in constant time. In

addition, BDDs are easy to manipulate. Any Boolean operation on two BDDs can be carried out in time proportional to the product of their size.

The size of a BDD can depend critically on the variable ordering. To find an optimal ordering is a co-NP-complete problem in itself [2], but a good heuristic for choosing an ordering is to locate dependent variables close to each other in the ordering. For a comprehensive introduction to BDDs and *branching programs* in general, we refer the reader to Bryant's original paper [2] and the books [13, 18].

3.3 BDD-Based Interactive Configuration

In the offline phase of BDD-based interactive configuration, we compile a BDD $\widetilde{S}(C)$ of the Boolean function representation $S'(C)$ of the solution space. The variable ordering of $\widetilde{S}(C)$ is identical to the ordering of the Boolean variables of $S'(C)$. $\widetilde{S}(C)$ can be compiled using a BDD version $\widetilde{\tau}$ of the function τ, where each Boolean operation is translated to its corresponding BDD operation

$$
\begin{aligned}
\widetilde{\tau}(x_i = v) &\equiv \text{BDD of } \tau(x_i = v) \\
\widetilde{\tau}(\varphi \wedge \psi) &\equiv \text{Op}_\wedge(\widetilde{\tau}(\varphi), \widetilde{\tau}(\psi)) \\
\widetilde{\tau}(\varphi \vee \psi) &\equiv \text{Op}_\vee(\widetilde{\tau}(\varphi), \widetilde{\tau}(\psi)) \\
\widetilde{\tau}(\neg\varphi) &\equiv \text{Op}_\neg(\widetilde{\tau}(\varphi)).
\end{aligned}
$$

In the base case, $\widetilde{\tau}(x_i = v)$ denotes a BDD of the Boolean function $\tau(x_i = v)$ as defined in Section 3.1. For each of the inductive cases, we first compile a BDD for each sub-expression and then perform the BDD operation corresponding to the Boolean operation on the sub-expressions. We have

$$
\widetilde{S}(C) \equiv \text{Op}_\wedge(\widetilde{\tau}(F_D), \widetilde{\tau}(f_1), \wedge, \widetilde{\tau}(f_m)).
$$

Due to the polynomial complexity of BDD-operations, the complexity of computing $\widetilde{S}(C)$ may be exponential in the size of C.

A version of the INTERACTIVE-CONFIGURATION procedure shown in Figure 3, where the Boolean functions are represented by BDDs, is used in the online phase. Each Boolean operation of this procedure is translated to its corresponding BDD operation. The response time of the procedure is determined by the complexity of performing a single iteration of the procedure. All sub-operations can be done in time linear in the size of *Sol* except VALID-ASS in Line 3. This procedure can be realized by a specialized BDD operation with worst-case complexity

$$
O\left(\sum_{i=1}^{n} |V_i| \|D_i|\right),
$$

where V_i denotes the nodes in *Sol* associated with BDD variables encoding the domain of variable x_i. As usual, D_i denotes the domain of x_i. For each value of each variable the procedure tracks whether the value is encoded by *Sol*. Due to the ordering of the BDD variables, for each variable x_i, this tracking can be constrained to the nodes V_i.

4. CONFIGIT SOFTWARE

Configit Product Modeller (Configit-PM) is a software product for interactive configuration developed and distributed by Configit software [4]. Configit-PM requires product models to be specified in its own strongly typed language. The language is simple but expressive enough to handle the product models occurring in real-life modeling applications. Configit-PM also has a compiler which first checks for the semantic correctness of the product model. If the semantics is valid then it creates a virtual table (VT) that contains all valid solutions of the product model. VTs are stored in XML format. Details about the variables in the product model including their domain sizes are stored in a header part of the VT file. The header is followed by a BDD derived data structure that represents the valid solutions of the product model. All the information necessary to configure a product is embedded in a single VT file.

Using Configit-PM, a preferred interface can be created to interact with the VT file for configuring a product. The Configit-PM also has a built-in simulator, PM-Viewer, which can be used to interact and configure products. PM-Viewer can be used to select a value for each variable in the product model. It also has an undo facility to go back and forth between selections. Implications of the user selections will be shown as forced selections. The user need not have to select values for all the variables. After selecting a value for zero or more variables, the user can request the PM-Viewer to complete the rest of the options automatically. The PM-Viewer can also give explanations for all the forced selections. When the user wishes to select an invalidated value of a variable, the PM-Viewer will show a list of previous choices to be undone to resolve the conflict.

5. EXPERIMENTAL WORK

In this section we present experiments we have carried out using Configit-PM. The results are listed in Table 1. First column lists a benchmark name. Four benchmarks were used in the experiments. Three successive columns list the time taken for generating the corresponding VT, the size of the VT, and, the number of valid configurations (#Solutions), respectively. Last column shows the average response time over 1000 random requests on the generated VT. Each request corresponds to an iteration of the INTERACTIVE-CONFIGURATION procedure. The Renault benchmark represents a car configuration problem used in [1]. The PSR benchmark represents a power supply restoration (PSR) problem. Information about the PSR problem is available in [16]. PC is a benchmark distributed by Configit [4] along with Configit-PM. It represents a personal computer configuration problem. Parity represents a parity learning problem as a configuration instance. Information about this problem is available in [7]. The results show that even though a problem may take long time to compile in the offline phase, it may result in a small VT that gives very short response times in the online phase. This is in particular true for the Renault benchmark. Also notice that VTs due to the symbolic encoding may represent large solution spaces very compactly.

Benchmark	Virtual Table			Average Response Time (sec)
	Time (sec)	Size (KB)	#Solutions	
Renault	460.00	1292	2.8×10^{12}	0.127
PSR	0.38	37	7.7×10^9	0.001
PC	0.89	24	1.1×10^6	0.075
Parity	30.00	1219	198×10^6	0.096

Table 1. Experimental results on four benchmark problems.

6. RELATED WORK

Related work can broadly be classified into search-methods based on Constraint Satisfaction Problems (CSPs) [9] and Boolean satisfiability formulations [14], and, compilation methods using data structures like acyclic constraint networks [5, 6] and Automata [1]. In this section we give an overview of some of them.

In [8], the authors propose a preprocessing method to convert a CSP into another one, having Backtrack-Free problem Representation (BFR). The BFR can then be used for interactive configuration. Unlike conventional preprocessing methods which add additional constraints and hence increase the size of the problem representation, their method does not add any additional constraints. Instead, they restrict the domain of variables, such that a BFR is obtained. The main drawback of this approach is that the BFR does not contain all valid solutions. They give a guarantee that their preprocessor will give an error message if all solutions are removed by it. Although the authors claim that deletion of some valid solutions by their preprocessing step is acceptable in many cases, hiding even a single valid solution from the user is questionable. Although their representation is backtrack-free like our BDD-based method, some solutions are lost and hence their method is not preferable in real-life product configuration systems. In [9], the authors use consistency methods to obtain *explanations* and *implications* for a configurator based on a CSP representation. They use the N-Queens problem to demonstrate their method. It is like other CSP-based search methods, which solve an intractable problem every time the user makes a request. In [18], the authors presented Minimal Synthesis Trees (MSTs), a data structure to compactly represent the set of all solutions in a CSP. It takes advantage of combining the consistency techniques with a decomposition and interchangeability idea. Unlike our approach, which generates worst case exponential-size BDDs, the MST is a polynomial-size structure. Operations on the MSTs, however, are of exponential time complexity while they are of polynomial complexity in our approach.

In [14], the authors present an approach to the configuration problem based on a Boolean Satisfiability (SAT) solver. They have developed a non-interactive configuration system (BIS) based on a new SAT solver designed by them. The BIS system was developed for a commercial car manufacturer. Although their technique can be extended to an interactive configuration method, it will not move the intractability of the configuration process into an offline activity.

The problem with search-based methods is that the intractability of the configuration problem is solved every time the user gives a request to the configurator. In case of compilation techniques, the advantage is that the compilation process may be intractable but once the valid solutions are compiled into an efficient data structure, the interaction process is efficient.

Acyclic constraint networks and the *Tree clustering algorithm* [5, 6] represent a CSP solution space in a more compact way, organizing it as a tree of solved sub-problems. The generated structure offers polynomial time guarantees for extracting a solution in the size of the generated structure. The size of the sub-problems, however, cannot be controlled for all instances and might lead to an exponential blow-up. The complexity of the original problem is dominated by the complexity of the sub-problems, which are exponential in both space and time. Nevertheless, this is one of the first compilation approaches used to solve CSP problems. There are efforts to cope with this exponential blow-up by additional compression using Cartesian product representation [12].

In [1], the authors present a method which compiles all valid solutions of a configuration problem into an automaton. After compiling the solutions into an automaton, functions required for interactive configuration, like implications, explanations, and valid-domain-calculations can be done efficiently. They also present a theoretical view of all the complexity issues involved in their approach. They show that all the tasks involved in an interactive configuration process are intractable in the worst case. They claim that intractability can be circumvented by compiling configuration problems into an automaton. That is, moving the intractable part of the problem into an offline compilation process. Technically this work is the closest one to our approach. They use automaton to represent valid configurations, where we use BDDs. The two approaches to two-phase interactive configuration may perform equally well. However, a major advantage of using BDDs is that this data structure has been studied intensely in formal verification for representing formal models of large systems [3, 19]. In particular, the variable ordering problem is well studied [13]. Furthermore a range of powerful software packages have been developed for manipulating BDDs [11, 15]. To our knowledge, the automata approach has not reached this level of maturity.

7. CONCLUSION

In this paper we have demonstrated how to solve the computationally hard interactive configuration problem by dividing it into two phases. First, in an offline phase, we compile the solution space of the problem to a Boolean domain using BDDs as underlying data structure. Second, if the resulting BDD is small enough, we achieve fast algorithms for interactive configuration in an online phase while providing user-friendly requirements such as completeness of inference.

The experimental results indicate that BDDs representing the solution space of real-world configuration problems are often small and enable very short response times in the online interactive configuration phase. Future work includes combining our approach with the state-of-the-art search-based techniques for solving the configuration problem. We also plan to generate more experimental results using benchmarks from different application domains.

REFERENCES

1. J. Amilhastre, H. Fargier, P.Marquis: Consistency restoration and explanations in dynamic CSPs—application to configuration. Artificial Intelligence, 135(1-2), 199-234, 2002

2. R. Bryant: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Transactions on Computers. Vol. C-35, No. 8, 677-691, 1986.

3. J. R. Burch, E. M. Clarke, K. McMillan: Symbolic model checking: 10^{20} states and beyond. In Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science, pages 428–439, 1990.

4. Configit website : www.configit-software.com

5. R. Dechter, J. Pearl: Network-based heuristics for constraint-satisfaction problems. Artificial Intelligence. Vol. 34 No. 1, 1-38, 1987

6. R. Dechter, J. Pearl: Tree-Clustering Schemes for Constraint-Processing. Artificial Intelligence. Vol. 38(3), 353-366, 1989

7. ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/contributed/crawford/README

8. E.C.Freuder, T.Carchrae, J.C.Beck: Satisfaction Guaranteed in Workshop on Configuration, Eighteenth International Joint Conference on Artificial Intelligence, 2003

9. E. C. Freuder, C. Likitvivatanavong, R. J. Wallace: Explanation and Implication for Configuration Problems in IJCAI-2001 Workshop on Configuration, Seattle, WA, August 2001, 2001

10. M.R. Garey, D.S. Johnson: Computers and Intractability-A Guide to the Theory of NP-Completeness, W H Freeman & Co, 1979

11. J. Lind-Nielsen: BuDDy - A Binary Decision Diagram Package. Technical Report IT-TR: 1999-028, Institute of Information Technology, Technical University of Denmark, 1999. `http://cs.it.dtu.dk/buddy`.

12. J. N. Madsen: Methods for Interactive Constraint Satisfaction. M.Sc. Thesis, Department of Computer Science, University of Copenhagen. 2003. http://www.diku.dk/forskning/performance-engineering/jeppe/

13. C. Meinel, T. Theobald: Algorithms and Data Structures in VLSI Design. Springer, 1998

14. C. Sinz, A. Kaiser, W. Küchlin: Formal methods for the validation of automotive product configuration data. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 17(1):75-97, January 2003. Special issue on configuration.

15. F. Somenzi: CUDD: Colorado university decision diagram package. `ftp://vlsi.colorado.edu/pub/`, 1996.

16. S. Thiebaux, M. O. Cordie: Supply restoration in power distribution systems a benchmark for planning under uncertainty. In Pre-Proceedings of the 6th European Conference on Planning (ECP-01), 85-96, 2001

17. I.Wegener: Branching Programs and Binary Decision Diagrams. Society for Industrial and Applied Mathematics (SIAM), 2000

18. R. Weigel, B. Faltings: Compiling constraint satisfaction problems. Artificial Intelligence 115(2): 257-287 (1999)

19. B. Yang, et.al,:A Performance Study of BDD-Based Model Checking. Proceedings of the Second International Conference on Formal Methods in Computer-Aided Design, 255 - 289, 1998

Reducing uncertainty with an augmented-reality based configuration system for furniture

Ralf Reichwald, Dominik Walcher

TUM Business School
Institut of Information, Organization and Management
Technische Universität München
+49 (0)89 289 248 26
walcher@wi.tum.de

Furniture can be seen as a customizable product par excellence. With the rise of the internet, many vendors have started to sell furniture online (within a mass customization strategy) offering the customers a huge variety of configuration possibilities. Most of these configuration systems are well elaborated and the configuration of the individual furniture can be done in an easy and playful way. However, one pivotal aspect, which is crucial for the customer's ability to make a valid purchase decision, is neglected by most vendors: "How does the furniture look in the future context?" In this paper an Augmented Reality based configuration system is introduced, which aims to overcome the limitation of current systems. With consideration of the theoretical construct "uncertainty", which occurs most often during the buying decision process, the advantages of the AFC (Augmented Furniture Client) are presented.

Significance: The AFC (Augmented Furniture Client), which is explained in this paper, aims to reduce uncertainty within the buying decision process of furniture. Augmented Reality (AR) has been used so far mainly in B-2-B transactions. The developed AR configuration system addresses the B-2-C market.

Keywords: Configuration System, Augmented Reality, Uncertainty, Customer Satisfaction

1. COMPETITIVE CONDITIONS

Competitive conditions on markets are changing globally. A clear trend towards globalization and corporate flexibility can be identified. Seller markets have shifted to buyer markets forcing companies to reorganize and invest enormously into programs for improving customer orientation ([1]). New information and communication technologies foster the customer's position by providing a high level of transparency concerning the tenders of different vendors as well as lower transaction- and switching-costs. Regarding to these great shifts only those companies can stay in businesses, which differ unequivocally from the competition by offering the customer added value. Individualization and customization are estimated in many times as appropriate approach. Therefore more and more vendors follow a mass customization strategy ([2]).

1.1 Mass Customization Strategy

The objective of mass customization is to deliver goods and services, which meet individual customer's needs with near mass production efficiency ([2]; [3]; [4]; [5]; [6]). This preposition means that individualized or personalized goods can be provided without the high cost surpluses (and, thus, price premiums) usually connected with (craft) customization. Therefore, mass customization is often connected closely with the capabilities offered by new manufacturing technologies (CIM, flexible manufacturing systems) reducing the trade-off between variety and productivity and hence enabling to decrease the additional production costs ([2]; [7]; [8]; [9]; [10]).

However, new flexible manufacturing systems are a necessary but not sufficient condition for successful mass customization. They have to be supplemented by infrastructures capable of handling the information flows and transaction costs connected with mass customization, which is characterized by a high intensity of information compared to mass production ([11]; [12]; [13]). This information intensity results from the fact that starting point of each customization are

specific customer needs. The supplier has to interact with each single customer and obtain specific information in order to define and translate the customer's needs and desires into a concrete product specification.

1.2 The customer's new role

By translating individual customer needs in a specific product, the customer is involved into the value creation of the supplier. The customer becomes a "co-producer" respectively "prosumer" ([16]). However, as the main part of the interaction with the customer takes place during the configuration and therefore the design of a customer specific product, it seems appropriate to call the customer rather a co-designer than a co-producer. Customer co-design describes a process that allows customers to express their product requirements and carry out product realization processes by mapping the requirements into the physical domain of the product ([17], [18], [19]). A product is co-designed between each single customer and the supplier.

The integration of the customer creates collaboration between the supplier and the customer, which supersedes the traditional value chain. Companies successfully pursuing mass customization build an integrated knowledge flow – that not only covers one transaction but uses information gathered during the fulfillment of a customer-specific order to improve the knowledge base of the whole company ([23]; [21]; [5]). During the whole process the interface between manufacturer and customer is crucial. Not only does it comprise the solution space of the production facilities, but it is also the design instrument for new and existing customers, the core communication tool, and supposed to be the main origin of customer loyalty (e.g. [24]; [25]; [26]).

Additionally, the interaction system is the prime instrument for reducing the user's costs arising from a principal-agent-constellation that is inevitable in mass customization. From the customer's perspective the co-design is connected with additional costs ([11]; [23]). Users often have no clear knowledge of what solution might correspond to their needs, sometimes they still have to find out what their needs are. As a result, customers may experience uncertainty during the buying decision process.

2. BUYING DECISION PROCESS AND UNCERTAINTY

In the following different kinds of buying decision processes and categories of the theoretical construct "uncertainty" are discussed in more detail.

2.1 Classification of buying decision processes

Buying decisions can be categorized concerning the level of cognitive control. Thus there are decisions with strong cognitive control (extensive and limited buying decisions) as well as buying decisions with a low level of cognitive control (habit- and impulse-purchases). Moreover emotions affect buying decisions, push cognitions and determine the customer's information-seeking-behaviour. Furthermore reactive processes, which are responsible for automatic reactions in certain situations, affect the buying decision. This mainly addresses the excitement of a buying-decision-situation, which is also influences by internal as well as external factors (e.g. personal attribute, pressure of time, etc.) ([27]; [28]).

Based on the mentioned factors all buying decisions can be categorized into four classes: extensive, limited, habitualized and impulsive – which are explained in more detail in the following ([27]):

Extensive Decisions (EDs):

EDs are coined by novelty and complexity. Moreover EDs need a high level of information and therefore they are long lasting in order to minimize the risk which is linked to the buying decision. This means that cognitive processes play a dominat role. Equally emotions can be seen as important factor, because they activate the cognitive processes. That's why the customer's attitude towards a product becomes manifest during the decision-making-process. This is caused by the novelty of the decision-situation.

Limited Decisions (LDs):

LDs are characterized by cognitive simplifications. In contrast to EDs novelty and innovation are missing. They are based mainly on knowledge and experience – the emotional factor is also absent. Information-reception and processing are pivotal elements, whereas internally stored information is evoked at first. Only if the internal information is not sufficient external sources of information are used. At this, mainly key-information is searched for. The customer's behaviour is affected by

the situation, which is not novel at all, but can be handled based on the customer's experiences. Copared to EDs reactive processes can be disregarded.

Habitualized Decisions (HDs)
 Similar to the LDs HDs are characterized by simplified decision-behaviour on an higher level. HDs take place reactively and automatically. Thus they can be found most often at repeated purchases, which can be conducted by the customer nearly without cognitive activities. The simplification is caused by familiarity-seeking and reduction of uncertainty, which explains product- and brand-loyalty to some extent.

Impulsive Decisions (IDs)
IDs are characterized by a high level of reactive decision-behaviour, which is accompanied by emotions and takes place automatically without many cogitations. Most of IDs can be found at not-planned purchases, which represent spontaneous need satisfaction at the point of sales.

2.2 Classification of the theoretical construct "uncertainty"

Uncertainty during a product-selection-process (design process) is based on the customer's imperfect subjective information. Contrary a decision made with certainty is based on perfect subjective information. Basically three different kinds of uncertainty within the decision making process can be differentiated ([27]); [29]):

Uncertainty concerning alternatives (UA)
At the beginning of a buying process the customer gathers a consideration-set of products, which seems to be relevant for the satisfaction of his needs. The lack of knowledge if the consideration set can be seen as actual base for the decision causes UA.

Uncertainty concerning image (UI)
The customer perceives the single product with its unique attributes. Based on these perceptions the customer develops certain images of the product. The question, if the perceived image concurs with the "real" attributes of the product, causes the UI.

Uncertainty concerning quality (UQ)
If the customer has identified the product image as sufficient nothingtheless UQ can happen. The perceived attributes are compared with the subjective desired attributes. Imponderabilities concerning the quality of the perceived attributes cause UQ.

Uncertainty can occur in two values: tolerable and un-tolerable. If the decision making process is dominated by uncertainty different reactions can be found:

- Cancellation of the process
- Iteration of passed steps
- Continuation of the decision process

If one of the three mentioned uncertainties isn't tolerable the decision process is cancelled or passed steps are repeated. Only if all uncertainties are tolerable the process is continued and a decision can be made. However in some cases the process is continued even with un-tolerable uncertainties. Here the perceived uncertainty is superseded. The repetition of passed steps is most often coined by the attempt to enhance the information base ([27]); [29]).

3. FURNITURE

Furniture can be seen as a customizable product par excellence. Based on modular product architectures and craft customization, furniture is offered as a customizable solution in many markets. The demand to customize furniture is very

much influenced by the planned locus of use, i.e. the position in an office or apartment where the product shall be placed. The future location of the furniture not only determines the measurements, but the customer's individual preferences also affect design and functionality [(30)].

With the rise of the internet, many vendors have started to sell furniture online within a mass customization strategy offering the customers a huge variety of configuration possibilities. Most of these configuration systems are well elaborated and the configuration of the individual furniture can be done in an easy and playful way. However, one pivotal aspect, which is crucial for the customer's ability to make a valid purchase decision, is neglected by most vendors: "How does the furniture look in the future context?" Only a few configuration systems offer "environmental functions" to simulate this future context in a virtual way, which is most often difficult to handle and time consuming. Moreover the result resembles the real situation to a minor degree and already existing pieces of furniture, to which the new article is supposed to accord with, can't be displayed at all.

The purchase of a piece of furniture, which is not a every-day event, can be seen as an extensive buying decision([29]). The process is quite complex and high level of imagination is needed. Most often the customers knows what kind of furniture is needed (e.g. a coffee table), but shape, colour, material, etc. are not decided at the beginning of the purchasing phase. Because of the aesthetical demand, which is connected with furniture, emotions play an important role. Last but not least the ignorance how the new piece of furniture will look in the new context can cause a high level of uncertainty.

As mentioned above different kinds of uncertainties occur within a buying process. The customer is able to see the product attributes, make changes and customize the piece of furniture with the help of an online configuration system. To imagine how the piece of furniture will look in the real living room is hardly feasible for most customers. Thus, the uncertainty concerning the image (UI) is addressed when buying furniture.

4. AUGMENTED FURNITURE CLIENT

In summer term 2003 an interdisciplinary project of the TUM Business School and the computer science department of TUM started, which was aimed to overcome the limitations of current furniture configuration systems. In cooperation with the Munich based start-up company "Augmented Solutions" (http://www.ar-solutions.de/as/index.htm) three students developed an Augmented Reality (AR) based configuration system – the AR Furniture Client (AFC).

4.1 Augmented Reality

"Augmented Reality (AR) is a newly emerging technology by which a user's view of the real world is augmented with additional information from a computer model ([31])."

Thus, AR means the overlay of virtual objects on the real environment. The "real world" is enhanced with virtual information by the means of 3D objects, text, graphics, etc. AR can be seen as the inversion of Virtual Realty (VR). While VR pretends a complete virtual world, AR inserts only specific virtual objects into the "real world". For that reason the user mostly has to wear special glasses, which enables the overlaying process. Most often a clear distinction between AR and VR is difficult - "interim worlds" come into existence. Figure 1 shows a classification of different "worlds" on the Reality-Virtuality-Continuum. Research on the psychological and cognitive impacts of VR is quite advanced whereas findings about AR are kind of rare ([31]).
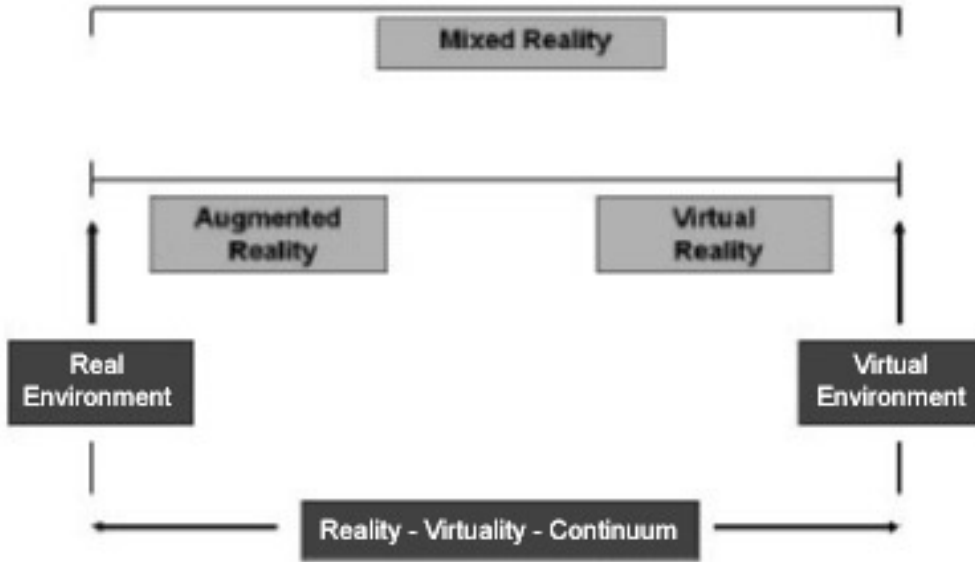
Figure 1. Reality-Virtuality-Continuum [(32)].

4.2 Functionality

Essential for Augmented Reality is image recognition. Based on image data, which is gathered by a camera, the system is able to identify information about the environment, such as position and movement of the user. The most common system is the so called "marker-recognition", which is also used by the AFC. Thus, a special pattern (see figure 2) is printed out and placed in the room. The AR-System recognizes the marker with the help of a special image recognition software-solution and calculates the direction of the user's cover bracket, which allows the overlay of virtual objects in the right angel and perspective. Moreover all movements of the user are registered in real time transforming the virtual objects in that way, that the user always experiences them as being static ([33]; [34]; [35]; [36])
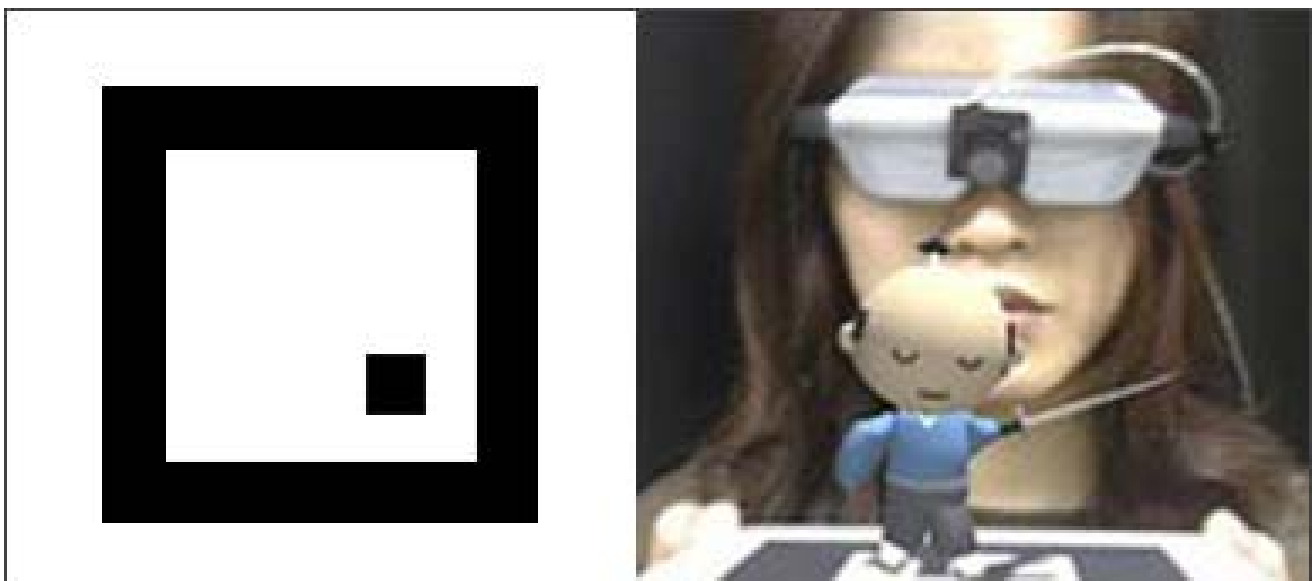

Figure 2. Marker-Outline (on the left), Virtual object on marker (on the right)

4.3 Input / Output Devices

The Head-Mounted-Display (HMD) is the most important output device. Using a so called "video-see-through" HMD the user sees the environment on two special displays. The picture on the displays is shot by a camera, which is fixed right in between. Thus the user doesn't see the "real" environment, but only a video-image. The video signal is transferred through a computer (main elements: scene generator and video mixer). Using the marker and the image recognition software the AR-software is able to enhance (augment) the scene with virtual objects, which are shown on the displays. For the user the imagination of <u>one</u> picture occurs ([31]).
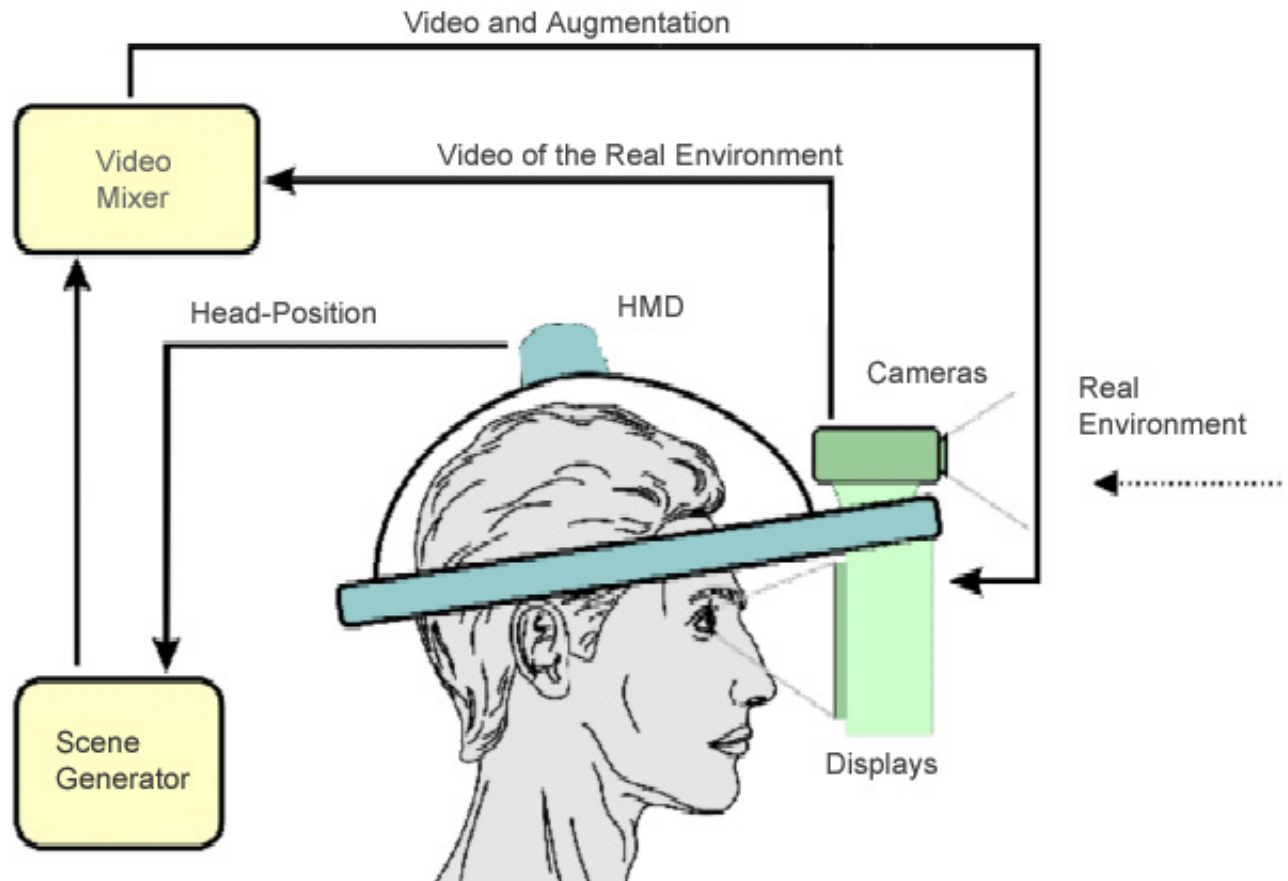

Figure 3. Augmentation with a "video-see-through" HMD ([31]).

4.4 Applications

Basically the AFC can be used in two different ways: Service assistant supported by the AFC (1) and the AFC as home application (2). In the following the two scenarios are explained in more detail with the help of the "Mortensen-Scenario":

Mr. and Mrs. Mortensen wants to purchase a new coffee table for their living room. Within the last weeks they have visited several furniture shops and have bought many magazines and catalogues. They could not make any decision because they were not able to imagine how the offered tables would look into their living room and if there's (aesthetical) harmony with the other pieces of furniture. So far the Mortensens have had a peaceful marriage, but not being able to decide unanimously, was a real test bench for their relationship! Finally they found the advertisement of IkeAR .This new furniture dealer is specialized in services around furniture configuration and visualization. IkeAR offers the "Augmented Furniture Client" and promises to have at least two solutions to cope with the Mortensens' problem:

1. Service assistant supported by the AFC

Mrs. Mortensen calls the IkeAR office and makes an appointment for the upcoming Saturday morning. A well dressed IkeAR agent appears equipped with a laptop, three HMDs and a marker printed on a regular DINA4 paper. Mr. Mortensen shows where the new table is supposed to be placed. The agent positions the marker. Afterwards he shows

the Mortensens a collection of several coffee tables on the laptop. The Mortensens select 5 different tables and put on the HMDs. Immediately the first selected table appears right on the marker and can be eyed from every side and perspective. By and by all tables are visualised in this manner and finally one design is selected by Mr. and Mrs. Mortensen. They like the shape but to harmonize perfectly with the other pieces of furniture in the living room the table has to be more narrow and green. The agent changes the virtual model of the table with two mouse-clicks. The Mortensens are satisfied, agree and order. Seven days later the table is shipped to the Mortensens, who are very content with the "self-designed" item.

2. The AFC as home application

The Mortensens can download a marker on the IkeAR website and print it out. They position it on the placein their living room where the table is supposed to be. Now the make pictures with their digital camera from different positions –the only thing they have to take care of is that the marker can be seen on all pictures. On the IkeAR website the pictures can be uploaded (see figure 4).After the upload the first picture of their living room appears. On a huge database thousands of different pieces of furniture are available. The Mortensens choose the coffee table entries and observe them on a preview window. They decide for a design and position it with one mouse-click right into the picture of their living room in the right size and perspective. With the help of different tools the table can be moved and turned to find the optimal position. The Mortensens choose another picture of their living room and the virtual model of the table appears as they have positioned it before in the new scene. They try another eight tables in all living room pictures, but finally decide for the first table. They slightly change the size of the table and select from a wide collection of surface materials the oak-grain and color it blue. They are satisfied and feel certain with their buying decision, because the AFC context sensitive visualisation supports their imagination. The Mortensens order the configured table online. Seven days later the table is shipped to the Mortensens.

Figure 4. Service assistant supported by the AFC (on the left), the AFC as home application (on the right)

4.5 Conclusion

As mentioned the uncertainty concerning the image (UI) is addressed when buying furniture. This uncertainty can be abolished if sufficient additional information is provided. The AFC offers the customer a high level of visual information supporting the imagination of how the piece of furniture will look in the future context. Therefore the AFC exceeds current furniture configuration systems, which have no or only rudimental environmental functionalities. It's argued that in situations where the customer experiences un-tolerable uncertainty and tends to cancel the purchasing process or to start again with the information-collection phase the usage of the AFC can decrease this uncertainty enabling the customer to continue and finally make the decision.

5. STUDY AND AWARD

5.1 Exploratory Study

In October 2003 the AFC home application was presented to the expert audience at the 2nd international congress on mass customization and personalization (MCPC'03) in Munich (http://www.mcpc2003.com). Over 3 days about 100 interested visitors tested the AFC-home-application-system at the stand. In not standardized discussions the participants were asked central questions concerning usability, reduction of uncertainty and relevance.

The qualitative statements were documented with headwords and analysed after the congress.
First of all there was great accordance of the relevance. Most participants confirmed that the system seems to be suitable for the reduction of uncertainty within the designing process. The majority of interviewed experts criticized the insufficiency of the current furniture configuration systems and the missing context sensitive visualization. Statements like "I wished I had such a tool when I bought my wardrobe last month!" were given quite often.

Some of the experts have heard of Augmented Reality, but never in the context of a B-2-C configuration system. Several times the difference to the well-known project ARVIKA (http://www.arvika.de), which was funded by the German ministry of research and education, was questioned. However the solutions and technical systems, which were researched within the ARVIKA project, focused on B-2-B applications in the branches automobiles, aeroplanes and power plants.

Speaking about the usability of the AFC-system most experts had several suggestions for improvements. The range of statements reached from very well usable to hardly operatable. Especially the levels of freedom could be enlarged as well as positioning, moving and turning functionalities.

Finally, the experts' overall satisfaction with the tool can be constituted as well. The students were encouraged by the majority of the participants to continue the development.

5.2 Award

In November and December 2003 the system was improved by the students with consideration of the gathered suggestions. The revised concept of the prototype was submitted to the international idea-competition "Innovation Award 2004", funded by a prominent German Consultancy and one of the largest German (business) newspaper-agencies. In March 2004 the winners were awarded at the Hannover based trade show CeBit. The winners were not published in advance. Thus the delight of the AFC team, which was invited to the CeBit, was tremendous when they were decorated with the first prize!

6. REFERENCES

1. Picot, A., Reichwald, R., Ralf, W. (2003). Die grenzenlose Unternehmung, Gabler, Wiesbaden.

2. Pine, J.B. (1993). Mass customization, Boston.

3. Duray, R., et al. (2000). Approaches to mass customization: configurations and empirical validation, in: Journal of Operations Managements, pp. 605-625.

4. Tseng, M., Jiao, J.(2001). Mass customization, in: Handbook of Industrial Engineering, edited by G. Salvendy, New York, pp. 684-709.

5. Piller, F.T. (2003). Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter, Gabler, Wiesbaden

6. Tseng, M., Piller, F.T. (2003). The customer centric enterprise. An integrative overview on this book, in: The customer centric enterprise: advances in mass customization and personalization, edited by M. Tseng, and F.T. Piller, Berlin et al., pp. 3-16.

7. Kotha, S. (1995). Mass customization: implementing the emerging paradigm for competitive advantage, in: Strategic Management Journal, pp. 21-42.

8. Anderson, D.M. (1997). Agile product development for mass customization, Chicago.

9. Victor, B., Boynton, A.C. (1998). Invented here, Boston.

10. Ahlström, P., Westbrook, R.(1999). Implications of mass customization for operations manage-ment, in: International Journal of Operations & Production Management, pp. 262-274.

11. Huffman, C., Kahn, B.E. (1998).Variety for sale: mass customization or mass confusion, in: Journal of Retailing, pp. 491-513.

12. Lee, S.C.-H., Barua, A., Whinston, A.B. (2000). The complementarity of mass customization and electronic commerce, in: Economics of Innova-tion & New Technology, pp. 81-110.

13. Reichwald, R., Piller, F., Moeslein, K. (2000): Information as a critical success factor for mass customization, in: Proceedings of the ASAC-IFSAM 2000 Conference, Montreal, Quebec.

14. Davis, S. (1987). Future perfect, Reading: Addison-Wesley.

15. Kotler, P. (1989). From mass marketing to mass customization, in: Planning Review, pp. 10-13.

16. Toffler, A. (1970). Future shock, New York.

17. Helander, M.G., Khalid, H.M. (1999). Customer needs in web-based do-it-yourself product design, in: Proceedings of the 10th Anniversary of M.Sc. Ergonomics International Conference, Lulea, Sweden.

18. Tseng, M., Du, X. (1998). Design by customers of mass customization products, in: CIRP Annals, pp. 103-106.

19. von Hippel, E. (1998). Economics of product development by users: the impact of "sticky" local information, in: Management Science, pp. 629-644.

20. Hibbard, J. (1999). Assembly online: the web is changing mass production into mass customization, in: Information Week, pp. 85f.

21. Zipkin, P. (2001). The limits of mass customization, in: Sloan Management Review, pp. 81-87.

22. Franke, N., Piller, F.T. (2003). Key research issues in user interaction with configuration toolkits, in: International Journal of Technology Management (IJTM), pp. 578-599.

23. Gilmore, J.H., Pine, J.B. (2000). Customization that counts, in: Markets of one, edited by J.H. Gil-more, and J.B. Pine, Boston, pp. vii-xxv.

24. Pine, J.B., Peppers, D., Rogers, M. (1995). Do you want to keep your customers forever?, in: Harvard Business Review, pp. 103-114.

25. Vandermerve, S. (2000). Personalization creates experiences, Keynote "First European Personalization Summit", London.

26. Riemer, K., Totz, C. (2003). The many faces of personalization - an integrative overview of mass customization and personalization, in: The customer centric enterprise: advances in mass customization and personalization, edited by M. Tseng, and F.T. Piller, Berlin et al.

27. Kroeber-Riel, W., Weinberg, P. (2003). Konsumentenverhalten, Franz Vahlen, München, 8. Auflage

28. Myers, J. H., Robertson, T. S. (1967). Consumer Behaviour and Marketing Management. Boston

29. Gerhard, A. (1995). Die Unsicherheit des Konsumenten bei der Kaufentscheidung, Wiesbaden

30. Steffen, D. (2000). Design als Produktsprache. Der „Offenbacher Ansatz" in Theorie und Praxis, Frankfurt a. M.

31. Reinhold, B., Klinker, G., Mizell, D.W. (1999). Augmented Reality. A K Peters Ltd.

32. Milgram, P., Takemura, H., Utsumi, A. and Kishino, F. (1994). "Augmented reality: a class of displays on the reality-virtuality continuum". In Proceedings of Telemanipulator and Telepresence Technologies.

33. Billinghurst, M. and Kato, H. (1999) Collaborative Mixed Reality. In Proceedings of International Symposium on Mixed Reality (ISMR '99). Mixed Reality--Merging Real and Virtual Worlds.

34. Billinghurst, M., Kato, H., Weghorst, S., Furness, T. A. (1999). A Mixed Reality 3D Conferencing Application (Technical Report R-99-1). Seattle: Human Interface Technology Laboratory, University of Washington.

35. Billinghurst, M., Kato, Hirokazu (1999). Real World Teleconferencing. Paper presented at Proceedings of CHI '99, Conference Companion.

36. Billinghurst, M., Kato, Hirokazu (1999). Collaborative Mixed Reality: Research Results. (Technical Report R-99-12). Seattle, WA: Human Interface Technology Laboratory, University of Washington. Presented at the Virtual Worlds Consortium.

TOWARDS A MODEL-BASED SOFTWARE PRODUCT LINE
FOR SMART CARDS

Stéphane Bonnet[1,2], Olivier Potonniée[1]

[1] Gemplus, Software Research Lab,
La Vigie, ZI Athelia,
La Ciotat (France)
http://www.gemplus.com/smart/
Email: {stephane.bonnet, olivier.potonniee}@gemplus.com

[2] Laboratoire d'Informatique Fondamentale de Lille,
University of Lille
Lille, France

Smart card configuration is a complex process, which has an important impact on the product price. Realizing it efficiently and with minimum development effort is a major challenge for smart card manufacturers. This paper exposes the motivations for applying modeling techniques to Software Product Line (SPL), in order to improve the smart card configuration and personalization processes. Our approach consists in modeling products and their variability. We illustrate the approach with the description of a model-based framework for card software final personalization

1. INTRODUCTION

Similar in size to today's plastic payment card, smart cards embed a microprocessor or a memory chip that, when coupled with a reader, has the processing power to serve many different applications. In particular, microprocessor cards can add, delete and manipulate information in their memory: they contain both data and the software to process it. As they provide data portability and security, smart cards are typically used as payment cards, or as identification modules plugged in mobile phones.

The very personal nature of smart cards makes their configuration a complex multi-level process, including a batch-specific configuration for each card issuer request, and a card by card individualization for the final card holders. For efficiency and homogeneity issues, we propose to use model-driven engineering to formalize the whole configuration life-cycle, and automate the configuration processes.

Model engineering is the emerging way of designing information systems and conducting software engineering projects. The switch from the programming to the modeling paradigm is meant to increase both productivity and quality of software engineering. Models enable the description of systems both with a higher level of abstraction, and with an improved methodology, for example by isolating different aspects of system as explained in the previous section.

We are experimenting an approach inspired by the Model-Driven Architecture (MDA), in which both products and their variability are modeled, refined, and finally projected on issuer- or user-specific target cards [i] [ii]. Section 2 presents smart cards and their configuration lifecycle. In section 3 we provide a very light overview of the current modeling landscape, before introducing our model-driven Software Product Line (SPL) approach for configuring smart cards [iii]. Finally, we describe in section 4 a framework for card software personalization, which is one of the last steps of the whole configuration life cycle.

2. SMART CARDS AND PRODUCTION PROCESS

In the past ten years, smart cards have essentially been used in the banking and telephony areas. The information age has progressively introduced new privacy and security issues that have called for advanced smart card applications. Smart cards are now playing an active role in three main contexts:

- Mobile telecommunications: in GSM networks, the smart card that is inserted in the mobile handsets secures the subscriber authentication. It also allows the storage of personal information and preferences of the subscribers, and enables advanced mobile value-added services.
- Commercial applications: the card stores information, money, applications which are used for banking, payment, loyalty, access control, etc.
- Information technology: the move of government and healthcare organizations towards storing and releasing information via networks has expanded the scope of uses for a smart card. In this context, the role of the smart card essentially relates to secure authentication, encryption of sensible data, storage of certificates, credentials, etc.

Most applications require a common hardware and basic software layer, on top of which domain specific applications are installed, in a configuration specific to the card issuer (e.g. the bank, or the mobile telecom operator). In addition, each card is personalized for its end-user, the card holder. This layered configuration process, illustrated in Figure 1, is an important specificity of the smart card business.
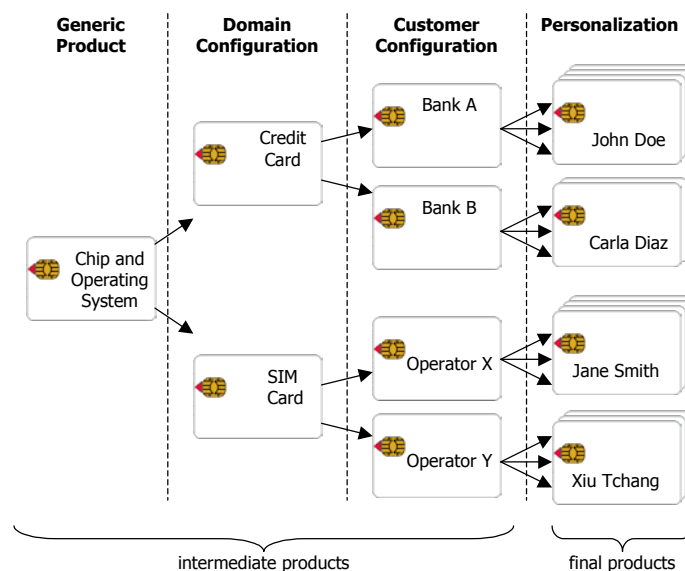

Fig. 1.: smart card configuration process

Smart cards are made of hardware and software, which both have specific production constraints. The hardware production consists in manufacturing chips with the appropriate ROM burned-in, assembling the chip and the plastic card, writing the application and customer configuration in the card programmable memory, and personalizing the card graphics. The card software engineering consists in producing all the software to embed, structuring its accompanying data, and producing the off-card software that will control the embedded configuration. Smart card configuration is hence a multi-level process. In order to maximize their revenue, smart card providers have to:

- Reduce configuration-specific development to the minimum. Intermediate products should be configurable enough to minimize the development required to produce a refined configuration. A single core product should be usable in as many final products as possible.
- Minimize production delays. To achieve this objective, hardware and software configuration should be done in parallel: the hardware production process should be started before the software configuration is completed. This allows to keep hardware products generic as long as possible, and to wait for purchase orders to quickly apply customer configuration.

Both objectives encourage to maximize reusability and product configurability.

3. MODELING THE SMART CARD CONFIGURATION LIFE-CYCLE

3.1 Model engineering: from contemplative models to MDA

An important paradigm change is taking place in software engineering, as modeling-based techniques are arousing more and more interest [iv] [v]. Models are abstraction of systems, they enable the representation of aspects of systems without engineering relative constraints. For a long time, their primary purpose has been to aid understanding and communication, but the trend today is to consider them as primary entities, from which refinement and projections are automated.

As domains, preoccupations, roles are heterogeneous, it is impossible to imagine a universal modeling language that would suit all the very different expectations of each practitioner. Therefore, models must be associated with precise semantics, which frees their interpretation from any ambiguity. A meta-model is a language to write models: it provides means to define the concepts used to create models. On a higher level, a meta-meta-model is a language to write meta-models. According to the Object Management Group (OMG), the Meta Object Facility (MOF) contains all the needed features to write and manipulate meta-models, and should therefore be considered as the universal meta-meta-model for software engineering [vi].

Most of the current research effort, both from industry and academia, is focused on Model-Driven Architecture (MDA). Created by the OMG to promote model engineering and leverage its existing modeling standards, MDA aims at providing an efficient framework for software and systems production. The specification basically proposes to organize models in two categories: Platform-Independent Models (PIM) and Platform-Specific Models (PSM). As the dependence on platforms may not always be a sufficient criterion to base a modeling framework on, the approach that consists in establishing a clear separation between different concerns can be the base foundation for further refinements of the MDA. Such refinements are part of the trend towards aspect-oriented modeling [vii].

For the real potential of model engineering to be reached, models must enable automated code generation and standardized projections. There is a lot of thinking among the research community about the different approaches for transforming models. A transformation between two models roughly consists in sets of rules, which establish correspondences between elements of the source model and elements of the target model [viii] [ix] [x]. The current trend is to consider transformations as models themselves, which implies the concepts used to define transformations must be defined in dedicated meta-models. In 2002, the OMG has posted a Request For Proposal (RFP) called MOF 2.0 Query/Views/Transformations [xi]. Its goal is to provide a unified transformation language.

3.2 A Model-driven SPL approach for smart card configuration

We propose to associate model engineering with the Software Product Line (SPL) approach, in order to improve the smart card configuration and personalization processes.

A SPL is a set of applications or systems relying on a common set of core software assets, and sharing a set of features satisfying the specific requirements of a given domain or problem [xii]. SPL-based approaches decompose the software engineering process in three interlaced activities: core assets development, products building, and organizational management. The derivation of multiple products from a common set of core assets is referred as variability. The ability to manage it efficiently at different stages of the development is essential for the SPL approach to reach its full potential. Variability points define where and when choices between different have to be made: they might thus relate to functional or technical issues. In the context of SPL, building a new product becomes more a matter of configuration and generation than one of creation. As explained in [xiii], model engineering particularly suits the SPL approach: it provides means to establish a clear distinction between the application model and the variability preoccupations, much like functional and technical preoccupations are kept separated in the MDA. The product derivation itself is realized through dedicated transformations.

Smart card products are made of several asset kinds, corresponding to each product maturity level. For instance, generic products are made of OS elements, whereas customer specific products reuse domain specific applications. To formalize their whole configuration life-cycle, we rely on model engineering techniques, and base our approach on a two level view of product configuration:

− the configuration description, which specifies a combination of potentially parameterized asset descriptors,
− the configuration projection, which is the materialization of a given configuration in a product.

The resulting process for a whole product configuration can be summarized as follows: from a set of asset descriptors, the designer uses a dedicated modeling tool to build a configuration. Each configuration can be enriched to form a more specific configuration. Any configuration can be projected on a specific target to become a product, or an intermediate product from which new derivations will be realized.
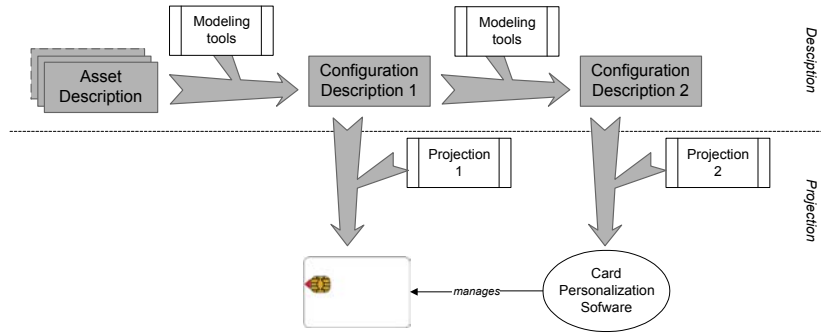
Fig. 2. Example of model-driven card configuration process

Figure 2 illustrates how model engineering applies in the smart card production context. We focus here on two of the last steps of the smart card software configuration process, with notably include the software personalization for final card holders:

− Designers are provided with modeling tools that allow them to combine and tailor assets descriptors (i.e., models) to produce the description of a specific configuration (i.e., another model). In the MDA world, this could be called the "platform-independent" layer.
− The first projection corresponds to the generation of the on-card software and data configuration. It includes the generation of source code, make files, and input data so that the industrial tool-chain can produce the desired result.
− The second projection generates on- and off-card code dedicated to the personalization of a specific product (i.e., an on-card application).

The real smart card production process includes more steps and transformations. But it is not our goal here to elaborate here on hardware, graphical, or other software issues for example. In the next section, we are describing our personalization framework prototype, which comes within the scope of this global model-based SPL approach for smart card configuration.

4. A FRAMEWORK FOR SMART CARD PERSONALIZATION

Our goal is to automate as much as possible the successive model creations and transformations required to instantiate a user-specific configuration of a given application. The personalization process consists in the following steps, which stand no matter how personalization is implemented:

− Identifying which elements of the application are user-specific.
− Organizing a structured management of end-user information.
− Realizing the mapping between the application adaptable elements and user relevant data.

For privacy and security reasons, we have chosen to realize personalization though a combination of both on- and off-card execution code. The global idea of the framework is to tailor the original application to make it "personalizable", and then to combine it with end-user personal data to generate personalized version of the application. These two steps correspond to variability points:

− The variability point concerns the definition of the elements of the application should be personalized. This manual step is of the responsibility of the personalization designer (who is not necessarily the original application designer).
− The second variability point is a refinement of the first one: once an element has been declared "personalizable", it is ready to be automatically personalized according to external parameters, i.e., user personal information.

In order to realize the separation of concern, we want to clearly isolate the personalization code from the functional one, so that the original application developer does not have to handle it. We are therefore relying on an annotation-based approach that enables proper extensions of generic models with personalization information.

As the current goal of our work is primarily to demonstrate the feasibility of model engineering applied to the smart card context, we are working with a light modeling environment. This furthers the focus on the methodology itself, rather than on tools and specification compatibility issues. Figure 3 makes explicit the MDA-like structure of the framework: while the above layer can be considered as a set of PIMs, the middle layer is a projection of these models on a specific platform, in that case the Java environment.
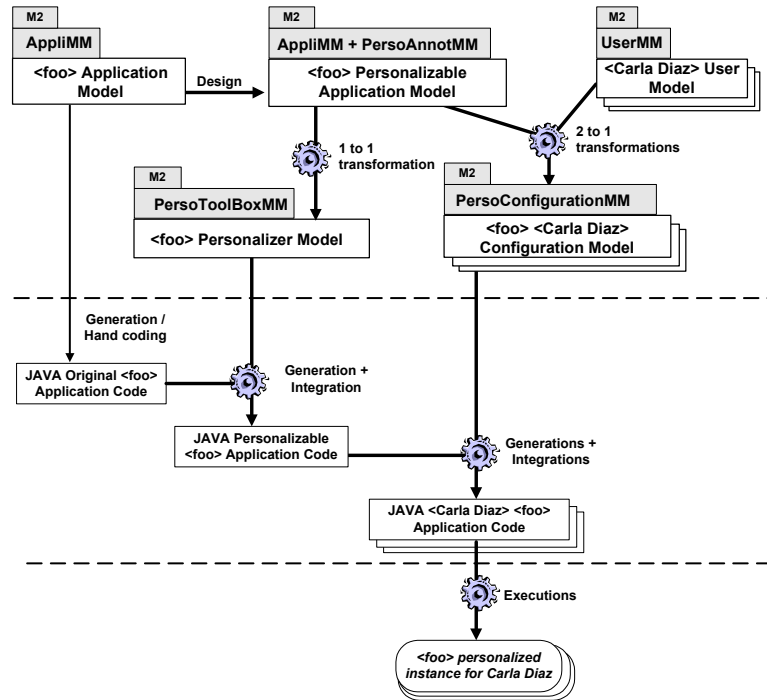
Fig. 3. The personalization framework

The framework contains two sub transformation chains resulting in two kinds of code modifications. As it is not user-specific, the transformation chain that makes the application code personalizable occurs only once for a given application. It consists in the following successive steps:

1. The `<foo>` application is designed with our `AppliMM` meta-model. The application designer does not need to integrate any personalization concern.
2. The enrichment of the `<foo>` application model with personalization information is realized using the concepts described in the `PersoAnnotMM` meta-model. Such personalization annotations can for example specify whether a class or an attribute can be customized and provide further details about it.
3. A "1 to 1" transformation takes the annotated model as an input, and returns a model of the personalization routines which must be added to the original application code. This `<foo> Personalizer Model` is based on the `PersoToolBoxMM` meta-model.
4. From this model are generated a set of Java classes and methods, which, when added to the original application code, make possible to configure the application state.

Once the application code is personalizable, user-specific configurations of the application can be instantiated after another succession of transformations:

1. Relevant data for the application personalization is structured as described by the `UserMM` meta-model in user models. For example, `<Carla Diaz> User Model` gathers information about Carla Diaz. More generally, each end-user is associated with one model.
2. A "2 to 1" transformation realizes the mapping between the configurable parts of the application and the user information to generate the `<foo> <Carla Diaz> Configuration Model`. This model describes the configuration of the application for Carla Diaz, using the `PersoConfigurationMM` meta-model.
3. The configuration model is used to generate user-specific code. This code is an application activator that uses the personalization routines generated by the first transformation chain.
4. The execution of the `<Carla Diaz> <foo> Application Code` results in a personalized instantiation of the application.

The design of the framework has been influenced by the chosen implementation, but with a bottom-up approach, we have managed to capitalize on some results, especially regarding transformations.

5. CONCLUSION

Smart card configuration is a complex multi-level process, including both hardware and software tailoring. The customization grain goes from huge clients such as banks or telecommunications operators, to individual card holders. We are naturally trying to maximize reusability and product configurability. Therefore, we are evaluating how modeling techniques can be used both to simplify and automate the configuration processes.

Even though models have been present in software development for quite a long time already, model engineering has only started few years ago to arouse a greater interest. The MDA from the OMG is undeniably the most spectacular illustration of this trend. In this paper, we have introduced an approach that consists in combining the Software Product Line approach with modeling techniques. To evaluate the feasibility of applying model engineering to the smart card context, we have prototyped a personalization framework, that has helped us to estimate the pertinence of our approach. The first results are encouraging us to continue the evaluation even though it is obvious practitioners might not be ready yet to reason at the model level. Therefore, rather than trying to change the existing in-place solutions, it seems more appropriate to focus on the next product generations.

It was not in the scope of this paper to precisely detail the smart card configuration life-cycle or to further describe the implementation details of the personalization framework, but some issues remain. In particular, the relative incompatibility of modeling tools, for example between different versions of the same specification. It is not relevant when prototyping, but it surely slows down the spreading of modeling engineering.

6. REFERENCES

???

i   MDA, Model Driven Architecture. OMG Specification (2001)

ii.   Model Driven Architecture (MDA) Guide Version 1.0.1, OMG Document (2003)

3.   P. Clements, L. Northrop, A Framework for Software Product Line Practice. Web document (2004)

iv.   J. Bézivin, From Object Composition to Model Transformation with the MDA. In IEEE proceedings of TOOLS USA (2001)

v.   J. Bézivin, S. Gérard, P.-A. Muller, L. Rioux, MDA components: Challenges and Opportunities. Workshop on meta-modeling for MDA, York, England (2003)

vi.   MOF, Meta-Object Facility. OMG Specification (2001)

vii.   Stephen J. Mellor, A Framework for Aspect-Oriented Modeling, The 4th AOSD Modeling With UML Workshop, (2003)

viii. A. Gerber, M.J. Lawley, K. Raymond, J. Steel, A. Wood. Transformation: The Missing Link of MDA. In Proceedings 1st International Conference on Graph Transformation (ICGT 2002), pp 90-105.

ix.   S. Sendall, W. Kozaczynski, Model transformation - The Heart and Soul of Model-Driven Software Development, In IEEE Software, Special Issue on Model Driven Software Development, pp. 42, (Sept/Oct 2003).

x.   K. Czarnecki, S. Helsen. Classification of Model Transformation Approaches, OOPSLA workshop on generative techniques in the context of MDA, Anaheim, USA (2003)

xi   MOF 2.0 Query/Views/Transformations RFP. OMG document. (2002)

12.   L. Bass, P. Clements, R. Kazman. Software Architecture in Practice. Addison-Wesley (1997)

xiii. S. Deelstra, M. Sinnema, J. van Gurp, J. Bosh, Model Driven Architecture as Approach to Manage Variability in Software Product Families, In Proceedings of the Workshop on Model Driven Architecture: Foundations and Applications (MDAFA 2003), University of Twente, June 2003.

PRODUCT KNOWLEDGE MANAGEMENT

Dr. Philipp Ackermann, Dominik Eichelberg

Perspectix AG
Hardturmstrasse 169
CH-8005 Zürich
Switzerland
Email: ackermann@perspectix.com, eichelberg@perspectix.com

The management of a product portfolio within an explicit product knowledge base is fundamental to support collaborative business processes within the corporate organization as well as between suppliers, marketing partners, and customers. To leverage business efficiency and simultaneously coordinate R&D, engineering, production as well as marketing and sales processes, companies require a unified view on product knowledge in order to better support and synchronize concurrent activities. Today's enterprises are moving away from unilateral, locally optimized views on product data (e.g., production-oriented in the ERP, engineering-oriented in the CAD) toward unified product models that foster editing of and access to product information for employees, partners, and customers throughout all business processes.

Significance: The networked economy generates an increasing demand on explicit product knowledge due to globalisa-tion, faster innovation cycles, as well as changing roles of suppliers and customers as co-designers and co-innovators.

Keywords: Product Knowledge Base, Collaborative Editing, Validation by Simulation.

1. PRODUCT KNOWLEDGE AS A CORPORATE RESOURCE

Markets are shifting from inflexible mass production and unprofitable design-to-order models to individually configurable product systems based on assemble-to-order fulfillment. Thanks to sophisticated on-demand manufacturing processes and IT-enabled production planning, companies can mass-customize their offerings in order to meet individual customer needs. Product knowledge is gaining growing attention for engineering collaboration, design reuse, modularisation and configuration of products as means to enhance productivity and performance in developing, manufacturing and selling customized products. The importance of product and configuration know-how is growing due to

1. Increasing complexity
Variant configuration is influencing a growing amount of product and service elements and corresponding processes.

2. Decentral expertise
Urgency for know-how alignment between different departments, different product lines and distributed locations

3. Heterogeneous IT tools
ERP systems in production and logistics, CAD systems in R&D, and SFA/CRM tools in sales define and depend on product data and configuration logic

The coordination and integration of relevant product information throughout the whole company is typically a task of the product management. In mass customization approaches, the management of product complexity and product variety is typically modeled within a product configurator [1, 2, 3]. We will summarize technical approaches and organisational experiences gained by developing and applying the P'X5 software suite [4] as a product configurator and authoring environment. The paper focuses on
*   conceptualizing products within an explicit modeling ontology based on Semantic Web (OWL, RDFS, XML) declarations
*   central coordination and integration of decentralized, federated product information using meta data definitions of a unified product model within a shared repository
*   model validation and quality insurance by simulating and testing dynamic product configurations

2. PRODUCT INFORMATION ARCHITECTURE

A product information architecture addresses the topic of organizing product information in a structured way (Figure 1). On a very basic level, it represents a unified product data model that is valid from several perspectives (dimensions, views) and includes technology considerations relating to data management. To generate information from data, a product information architecture must define role-based data access strategies that transform data structures into dynamic model simulation, vivid product representations, and meaningful user interaction. On a meta level, the semantics of the modeled problem domain are depicted by background theory, explicit conceptualization, design principles, as well as a common vocabulary.
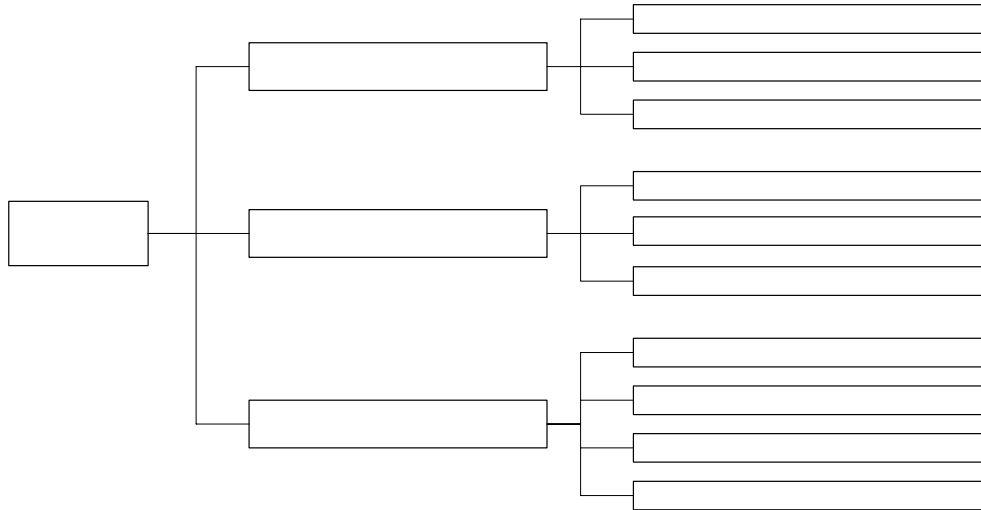

Figure 1. The information architecture includes data model, data access and modeling methodolgy

2.1 Unified Product Data Model

The goal of an explicit product data model is to represent products digitally and to exchange product information uniformly and efficiently. Product data are typically generated in a decentralized fashion within a heterogeneous IT infrastructure consisting of several applications such as CAD, PDM, PPS, ERP and even spreadsheets. The merging and linkage of such federated data via the standard XML format is a proven technology. The self-descriptive features of XML and its capacity to add semantics via meta data definitions support the central coordination and integration of decentralized product data.

An example of an XML-based modeling and integration platform is VCML (Visual Configuration Modeling Language). VCML is a language for expressing product information in a form that can be understood by a machine. It extends the syntactic interoperability of XML to a semantic level by precisely defining terms relevant to the domain of product modeling and product configuration.

Product configuration is a central problem in the product modeling domain, because it not only deals with fixed (configured) product structures but also with the potential structures of configurable products. The product data model must therefore reflect the dynamics and openness of modular products. A product model and its semantic meta model must promote the exchange of data by providing (meta) data on the data, and they must also facilitate machine processing of the product data.

2.2 Product Simulation

Products expressed as VCML models are not abstract ("dead") descriptions but valuable assets. Thanks to simulation by the P'X5 software, VCML models are executable and therefore comprehensible to humans due to the interactive responsive-ness of the product models. The P'X5 product simulator interprets the VCML descriptions and generates a virtual product representation with all the corresponding features. No procedural programming is needed in VCML modeling – the product configurator derives dynamics and constraints from the formal VCML declarations. The configurator software executes algorithms on the VCML data space such as expression calculation, rule-based reasoning with backtracking, and constraint solving of logical and geometric associations.

Information
Architecture

By combining the dynamics of the product simulation with interactive access, P'X5 software serves as an interactive product configurator. Users of the configurator can build their own product according to their individual requirements, and all the constraints of the VCML product (meta) model remain valid.

2.3 Product Modeling Ontology

An ontology constitutes a shared conceptualization of a specific domain. It embodies a certain world view of a domain and portrays a common understanding that can be communicated between people. If the ontology is expressed in a formal manner, it can even be processed by heterogeneous and distributed software applications.

VCML represents such a formal ontology. It is a synthesis of object-oriented, connection-based and rule-based product modeling approaches. Because the product component is the central concept, VCML is sometimes called a component-oriented modeling language. In VCML, the product modeling domain is conceived as a hierarchical description of important concepts in a type or class hierarchy, as a set of crucial conceptual properties, and as interrelationships between concepts.

The ontology and content of VCML identify the relevant concepts of the real world phenomena of a product. To create a product model according to a specific ontology, a consensual process among several people from different backgrounds is required. The purpose of an explicit product modeling ontology is to facilitate a common understanding among different people such as engineers with their CAD/CAM experience, production planers with a PPS/ERP background, marketing and sales people using CRM tools, IT engineers, etc. If all these people agree to accept an ontology, they can contribute to a unified product model.

2.4 Visual Configuration Modeling Language

The visual configuration modeling language (VCML) is a declarative modeling language that expresses generic product structures, dependencies and behavior. It serves as a XML-based integration middleware for product data. VCML applies descriptive representation formalisms that correspond to Semantic Web languages. The Semantic Web, initiated by the World Wide Web Consortium (W3C), is based on open standards such as XML, RDF and OWL. Figure 2 depicts the representation layers of VCML.

VCML	domain-specific semantics for product models	**Semantic Layer**
OWL	generic meta model / semantic web	
RFD schema	generalizations for classes and properties of RDF	**Data Layer**
RDF	data model for resources and relations	
XML schema	language for restricting XML data structures	**Syntax Layer**
XML	syntax for structured data	

Figure 2. The representation layers of a product modeling ontology

On the syntax level, the extensible markup language (XML) provides a syntactic representation of self-describing data structures. XML Schema is a language for restricting XML data structures. All VCML language expressions are defined by XML schemas.

The data layer is based on the resource description framework (RDF). RDF is a simple data model on top of XML for resources (objects) and the relations between them. RDF is itself expressed in XML. RDF Schema (RDFS) is a data model for describing classes and properties of RDF resources and has the capacity to represent inheritance along generalization hierarchies with "subClassOf" associations and range/domain constraints on properties. The vocabulary of a problem domain is expressed within RDFS.

In addition to RDFS, the OWL Web Ontology Language provides formal semantics so that computers can process XML-based content. OWL defines the restrictions on property cardinality, property domains and property types (e.g. transitive, symmetric). Additionally, OWL divides the RDF property into two disparate parts: the object domain and the data type domain. The object domain of "ObjectProperty" expresses the associations between two objects. The data type domain of "DatatypeProperty" relates objects to attribute values that belong to the XML Schema data types. Whereas OWL is a meta model for generic abstractions, VCML is domain-specific and reflects descriptive logic for generic product models. The ontology of VCML is shown as an OWL-like diagram in Figure 3.
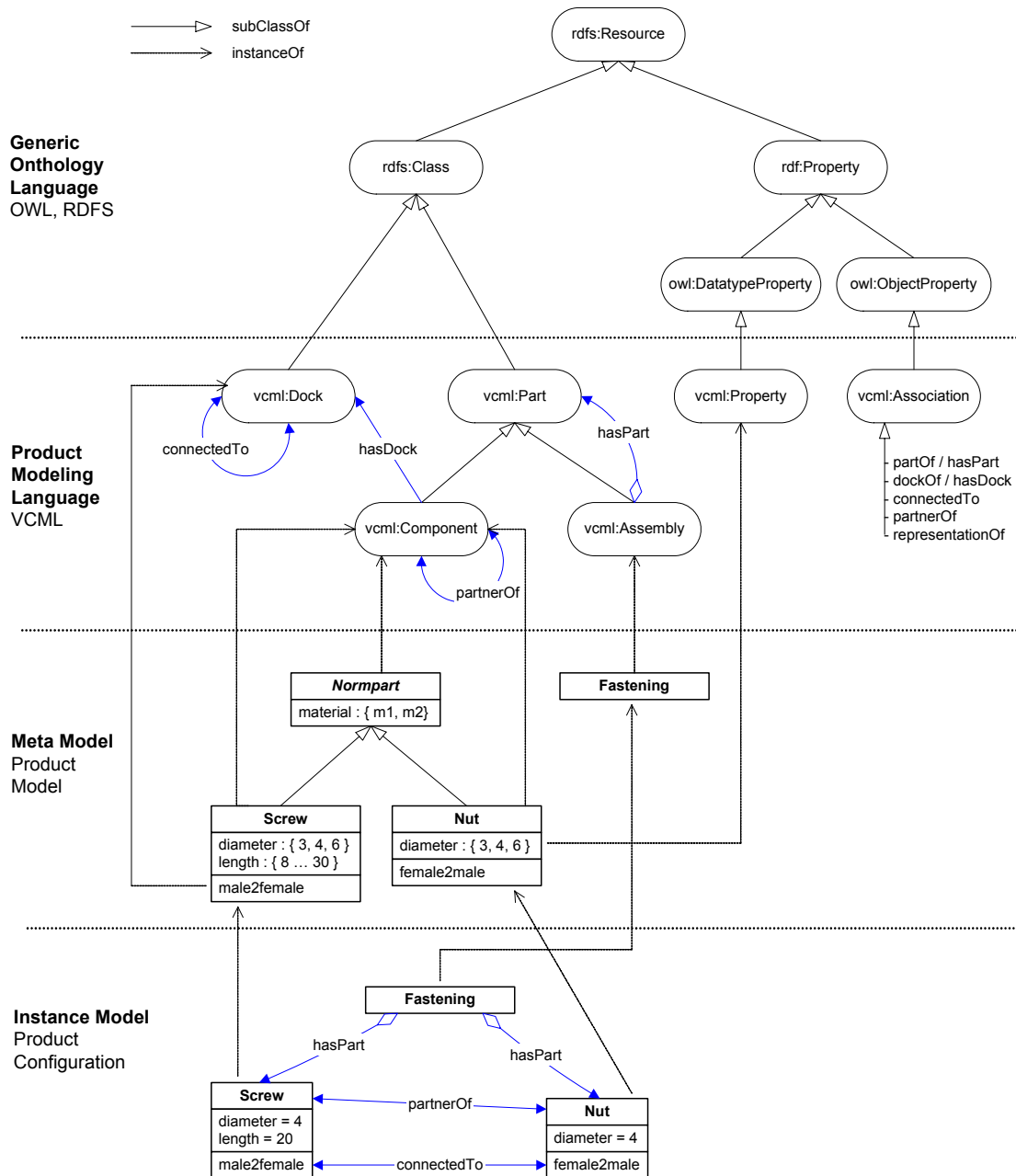
Figure 3. The modeling layers of a VCML product knowledge base

The main conceptual entities of VCML are

- Component: an atomic product element
- Assembly: an aggregation element used to group components and assemblies
- Part: product element as a super-type of component and assembly
- Property: an attribute with a domain
- Dock: a geometric interface or an attachment point for another dock

A product model consists of a structured breakdown of a real-world product line into corresponding entities that are classified by type systems. The coupling of such entities is expressed by associations:

- Specificational association: hasProperty
- Spatial association: hasDock
- Classificational association: subClassOf
- Compositional associations: connectedTo and partnerOf
- Aggregational association: hasPart
- Representational association: representationOf

- 160-

Association Type	Domain Type	Range Type	Cardi-nality	Inverse Association	Tran-sitive	Sym-metric
subClassOf	Type	Type	n:1	superClassOf	Yes	No
instanceOf	Instance	Type	n:1	isA	No	No
hasDock	Component	Dock	1:n	ownedBy	No	No
connectedTo	Dock	Dock	1:1	connectedTo	No	Yes
partnerOf	Component	Component	m:n	partnerOf	Yes	Yes
partOf	Part	Assembly	m:n	hasPart	Yes	No
hasProperty	Part	Property	1:n	ownedBy	No	No
representationOf	Representation	Part	n:1	representedBy	No	No

Table 1. Overview of the VCML association types and their properties

On the basis of the VCML expressiveness, product knowledge bases combine formal and spatial data such as product elements, classifications, assembly structures, bills of material (BoM), articles, labeling, pricing, colors, geometry, fitting conditions, etc. In VCML, the product model remains abstract and is therefore clearly separated from the representational structure used to display a model. With this distinct separation between model and view, it is possible to represent the same product model in several disparate representations such as schematic 2D, realistic 3D, sales-oriented article list, production-oriented BOMs, etc. Each representation type has its own viewer type. These viewers differ not only in the way they represent product information through specific structures and visualizations, but also in their distinct interaction modes [5].

3. COLLABORATIVE PRODUCT KNOWLEDGE MANAGEMENT

Editing and maintaining a unified product knowledge base at the overlap of innovation, marketing and fulfillment – e.g., in the product management process – is a great challenge. The purpose of a unified product model is:

- to harmonize product structures and processes on a generic meta-level
- to form a common vocabulary and understanding in product modeling
- to bridge mental and organizational boundaries in collaborative business
- to form a common vocabulary that reflects the product domain (and not the terms of a specific tool vendor)
- to foster the redesign / leverage / refactoring / restructuring / reuse of existing product solutions
- to foster the design of new products and product line extensions
- to provide a conceptual framework for the electronic exchange of product data between ERP, CAD, CRM, etc.
- to build an integration middleware that synchronizes tools which depend on product information and variant configuration logic
- to specify organizational responsibilities and release processes for product information
- to support the generation of digital product knowledge bases for product configurators

Collaborative engineering becomes collaborative business by involving marketing, sales, and service organizations into the product specification and innovation process. A unified product knowledge base needs therefore to integrate multidimensional requirements to product modeling:

- Engineering-oriented (geometry, material, parts, assemblies, constructive fittings)
- Production-oriented (scheduling, process rationalization)
- BoM-oriented (BoM structure, supplier-dependencies, available-to-promise calculation)
- Assembly-oriented (matings/fittings, reduction of assembly steps)
- Sales-oriented (portfolio communication, pricing, labeling)
- Service-oriented (delivery, on location assembly, testing, maintenance, repair, financing)
- Customer-oriented (usability, customizing, configuration)
- Cost-oriented (target costing, life-cycle costing)
- Ecology-oriented (sustainability)

3.1 Collaborative Authoring Environment

The generation and maintenance of a unified product model needs team work including specialists from product design, engineering, production, sales, and service. Product data is typically generated decentralized in diverse departments within a heterogeneous IT infrastructure consisting of several applications such as CAD, E-CAD, PDM, ERP, spreadsheets, technical documents editors, etc.. The reuse of existing product data, e.g., the integration of master data and graphical data into the product knowledge base, is fostered by the open, XML-based data schemas of VCML. On top of the master data, product logic in form of mathematical expressions,

constraints and rules are added. Due to the dynamic nature of product logic, product modeling is an iterative process consisting of data integration, logic definition, and validation.

To support short iteration cycles in the modeling process, the P'X5 authoring environment keeps the product knowledge base locally on the expert's workstation. The integration of additional product data, the modification of classification trees, the coding of logic rules are all developed and tested locally in a "controlled" sandbox. When the modeling expert reaches a certain consistent and validated milestone, he will check all modifications back into a central repository where the product knowledge base is kept under version control. Simultaneous work by distributed modeling team members is synchronized by committing to and updating from the shared repository.

Tool integration and team support of the P'X5 authoring environment is based on the Eclipse [6] platform. All projects in the workspace with all files and folders are managed by an integrated workbench (Figure 4) and are placed under history and version control with an associated team repository. Synchronisation with the repository works via LAN and via secure Internet access, therefore internal as well as external team members may collaborate on a unified product knowledge base.
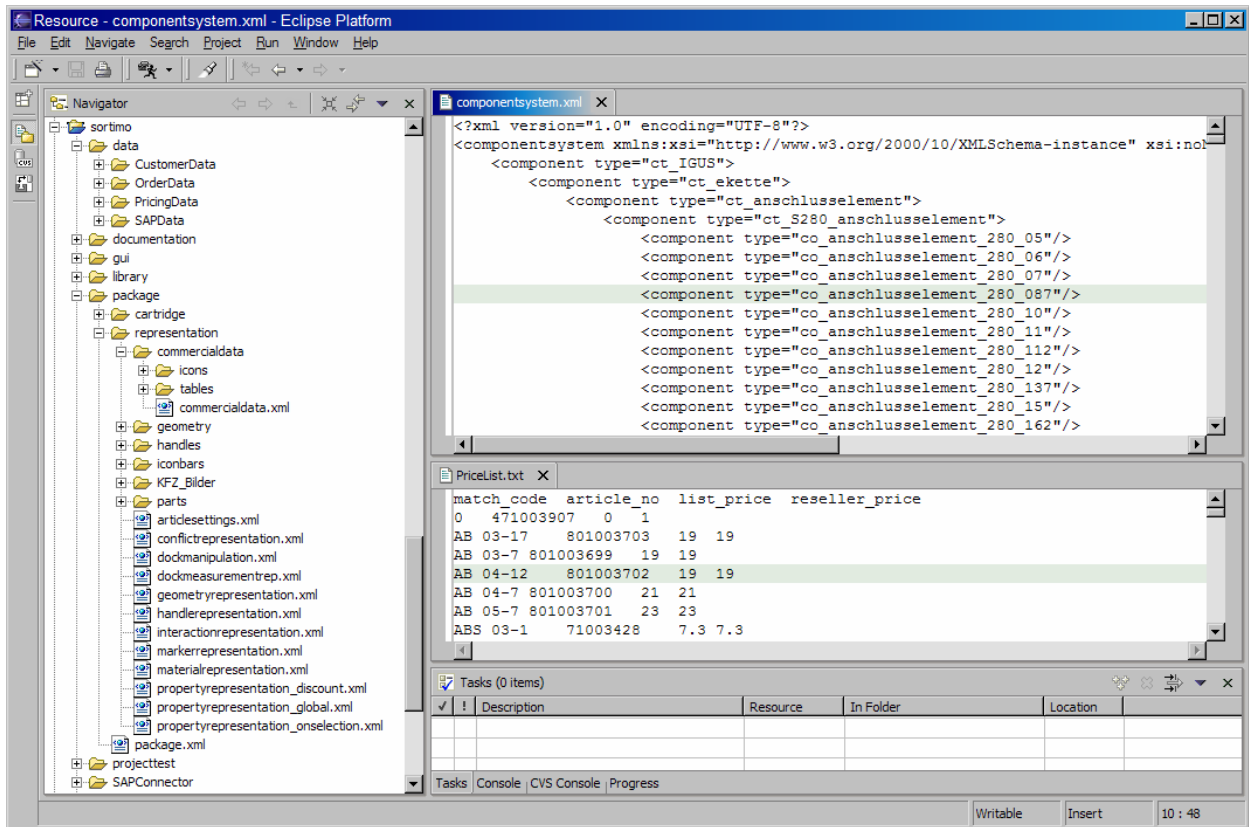

Figure 4. The workbench of the P'X5 product modeling environment provides tool integration and team collaboration

The workbench provides integrated browsers, editors and wizards for various aspects of the VCML-based product knowledge base. These browsers and editors hide the files and folders structure of a project and give access to symbols and their classifications which reflect the conceptual entities and associations of the VCML product modeling ontology. In Figure 5 on the left, a matrix view displays aggregational associations between the component classification and the assembly classification. On the right, a XML editor with schema support gives hints on VCML syntax and keywords while displaying the VCML structure in a table view.
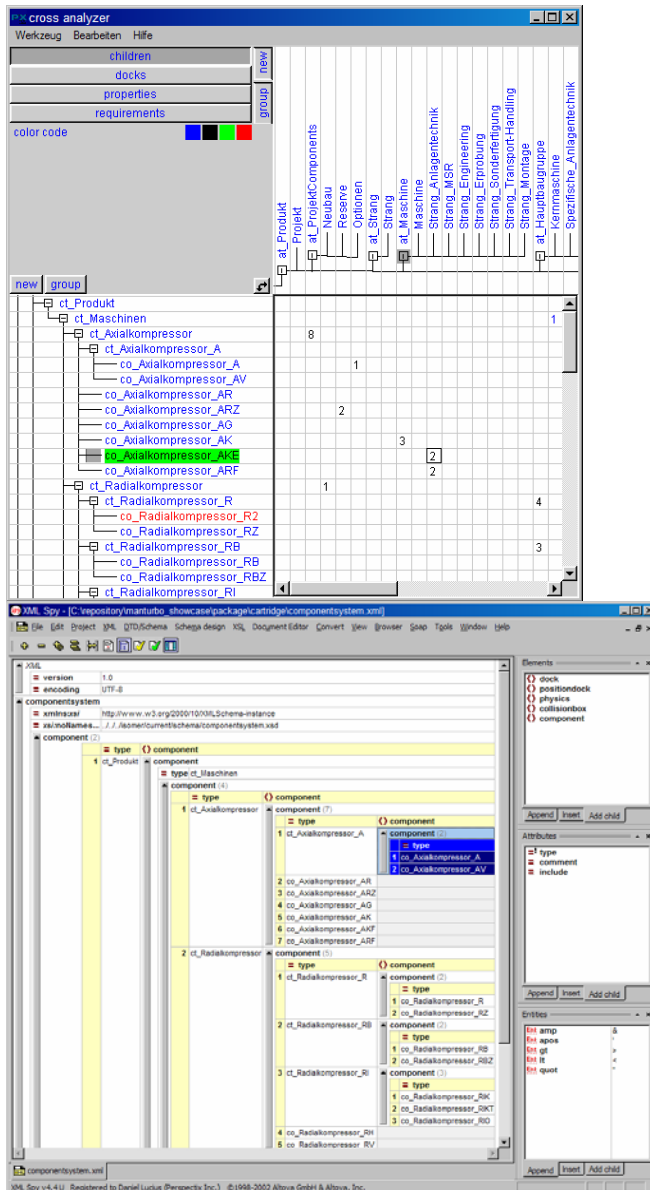
Figure 5. Browsers and editors of the P'X5 product modeling environment

3.2 Browsing, Editing, and Run-time Inspection

While developing a product knowledge base in an iterative process, a modeler is constantly switching between meta model and instance model (see Figure 3). On the meta level, product entities such as components, assemblies, and properties (characteristics) are imported or added and then structured in classification systems. Potential associations between these product entities are modeled on an abstract level by type assignments, compatibility matrices, mathematical expressions and logical rules. The classification systems define the inheritance hierarchy for associations. Additionally, associations heavily depend on concrete instances of product entities and their compositional and aggregational structuring. For example, rules may check the state of connected (partner) as well as composite (assembly) elements. Therefore, the P'X5 product modeling environment integrates its meta-data editors with run-time inspectors and graphical debuggers (Figure 6).
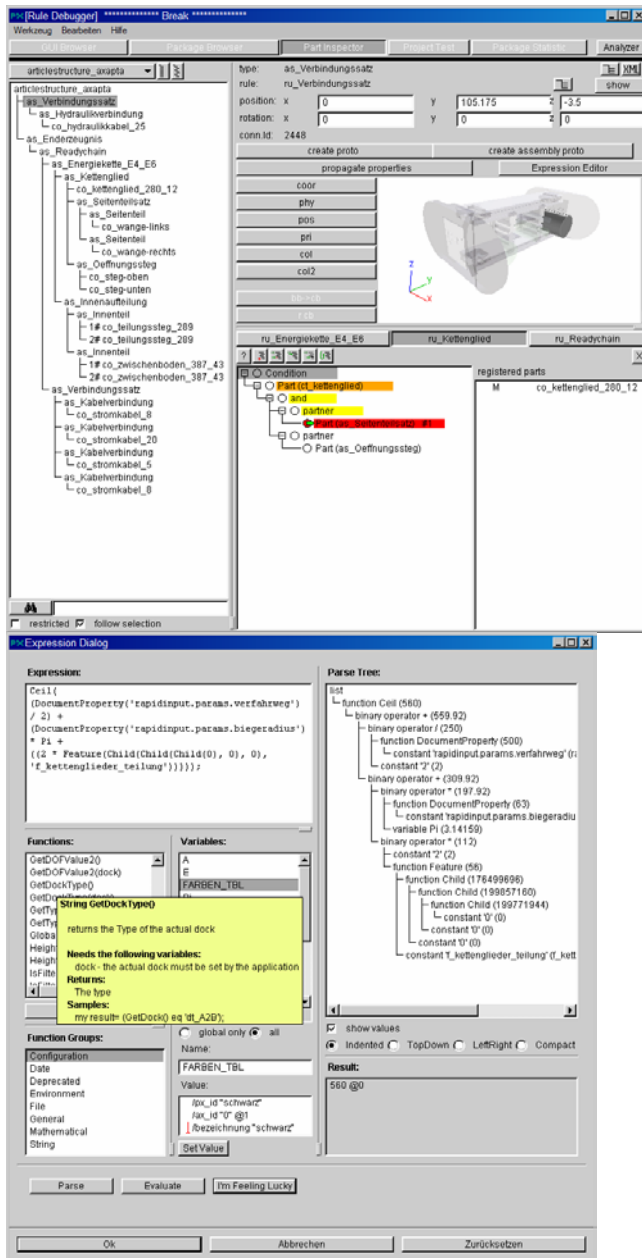
Figure 6. Run-time inspector with graphical rule debugger and expression editor

3.3 Validation by Automatic Product Simulation

While merging and linking federated product data, the quality management of product information and its configuration logic is fundamental in highly integrated and therefore heavily dependent IT systems. The central coordination and integration of decentralized product data benefits from the meta data definitions of a unified product model. The VCML schema definitions validate the consistency on a syntactical level as well as the referential integrity of all elements in the product knowledge base.

The correctness on a semantic level has to be validated by product-specific regression tests. These tests depend on dynamic product simulation whitin the configurator. Test scripts can iterate over product elements and dynamically execute operator sequences. The result of the operators' execution is validated against expected results. A modelling expert will not only define the meta data of his domain, but also the corresponding test scripts. These test scripts are collected within the product knowledge base and run automatically on certain events, e.g., before a check-in from a local workspace into the repository or on nightly tests on the repository server. A product manager "A" may redesign the article structure and pricing calculation scheme within a product knowledge base. If the product designer "B" has specified test scripts for the constructive assembly structure, the automatic testing assures that changes from the product manager "A" do not affect the technical

correctness. Of course, there will be no 100% assurance on the semantic level, but the test framework provides a tool infrastructure to constantly increase product data quality and logic consistency.

3.3 Product Modeling Patterns

While applying the introduced modeling methodology in several projects, typical modeling patterns emerged which proved their soundness in conceptualizing real-world product artefacts. These modelling patterns build a common vocabulary to discuss and document structural dependencies. The following modeling patterns are often used examples:

- Classificational patterns: type system, abstract type, property inheritance, dock inheritance
- Specificational patterns: discrete/continuos domain, preference value, conditional value/domain, mapped property
- Spatial patterns: collision box, parametric dock, parametric geometry
- Compositional patterns: symmetric dock, blocking dock, compositional property propagation
- Topological patterns: direct/indirect partner, conditional partner, partner pair/chain/ring/star/network
- Aggregational patterns: assembly, multi-structuring, volume grouping, aggregational property propagation

5. CONCLUSION

The modeling methodology in this paper is based on the experience of integrating engineering-oriented product data from CAD systems and manufactoring-oriented product information from ERP systems. Benefits from an explicit product information architecture with a modeling environment that supports team collaboration are

- Improving the management of complex and variant-rich product portfolios
- Expanding collaboration from engineering to a broader business focus by involving R&D, external suppliers, engineering, marketing and sales into the product specification and innovation process
- Unifying federated product data sources with a modeling ontology and an XML-based product knowledge base
- Validating the product knowledge base automatically within a product configurator by dynamic regression tests
- Providing user-friendly access to complex product models by a visual product configurator that integrates graphical CAD data and formal ERP data

6. REFERENCES

1. Asko Riitahuhta and Antti Pulkkinen, eds. (2001). Design for Configuration. Springer, Berlin.

2. Hans-Joachim Franke et al. (2002). Variantenmanagement in der Einzel- und Kleinserienfertigung. Hanser, Munich.

3. VDI (2001). Variantenvielfalt in Produkten und Prozessen: Erfahrungen, Methoden und Instrumente. VDI-Verlag, Düsseldorf.

4. Perspectix: Perspectix Home Page. http://www.perspectix.com/

5. Philipp Ackermann. Produktvisualisierung und Interaktions-Konzepte für Mass Customization; in: Piller, Schenk, Seelmann-Eggebert (eds): Tagungsband zur dritten deutschen Mass Customization Tagung, Frankfurt 2001.

6. Eclipse: Eclipse Home Page. http://www.eclipse.org/

PRODUCT CONFIGURATION SYSTEMS AND PRODUCTIVITY

Jørgen Lindgaard Pedersen, Kasper Edwards
First author is corresponding author

Department of Manufacturing Engineering and Management
Technical University of Denmark
Produktionstorvet
Bygning 423
DK - 2800 Kongens Lyngby, Denmark
Email:{jlp@ipl.dtu.dk}

Abstract

Twelve companies have been interviewed with the purpose to get information about technical, economic and organizational matters in respect of Product Configuration Systems (PCS). Combinations of qualitative interviews and quantitative scoring have been used in ranking expected and realized results from implementing PCS. The three highest aggregated scoring expected benefits are: 1) improved quality in specifications, 2) lower turnaround time, 3) less resource intensity. The PCS affects the quality of specifications in two ways: 1) validate configurations against a set of rules, 2) ensure all relevant information from the customer is collected. Using a PCS it is impossible to complete a configuration without all required information. Lower turnaround time is a direct consequence of automating the process of configuration a product. Using a product model is much more efficient than using direct labor in many functions.

1. INTRODUCTION[16]

A *Product Configuration System (PCS)* is an IT based system installed in a computer which can speed up time needed which a company shall spend to give a potential customer an answer about quality, price and time for a wanted good. Further the system can prepare such a tender much cheaper than a traditional tradesman can do it. Where such systems are integrated with production we shall talk about *Product Specification Systems (PSS)*.
In principle you can make a standard investment calculation of this system and do the same for the way in which you alternatively give tenders. Compare them and choose the system with the highest internal rate of return. Usually decision of investment in new technology will be made from a state of uncertainty. There exist *different types of uncertainty:*

- *Technological uncertainty:* will the system work as expected?

- *Commercial uncertainty:* what about prices for programs and services?

- *Social uncertainty:* can customers or competitors get sensitive information?

Uncertainty in respect of investment decision making shall be classified according to the following dimension:
1. *Objective uncertainty:* the character of the technology which can give results according with a constant probability distribution or be pure stochastic.

2. *Subjective uncertainty:* the decision maker does not know the precise outcome. He can know or guess a distribution of probability of a certain outcome. Alternatively he can make his decision in total fog.

In this paper we shall describe and analyse the following problematic:
1. The good to analyse – the tender. Demand and supply without a real market and a real price.

2. Some theory

---

3. The competition between traditional way to handle tenders and tenders delivered from PCS.

4. Expected costs and benefits from PCS in two case companies.

5. Realised costs and benefits from PCS in our two case companies.

6. Results from other companies

7. Discussion.

8. Conclusion.

2. THE MARKET FOR TENDERS

The goods we are looking at in this paper and in our project are characterized as goods which are *customized* according with wants and needs specific for individual customers. Usually the goods are *complex* in technical sense and very *expensive (indivisible).*

The production of such goods takes place according to an agreement in form of a contract between producer and buyer. Therefore it is an important activity to establish an order which takes place after a *tender* has been given from the sales division of the company. To do that it is a necessity to specify many different characteristics. That means the specification process can and in the *traditional way* will be a very knowledge and labour intensive process.

To analyse what happens about incentives to get and give tenders we have to notice that potential customers who apply to a company to get a tender of a wanted product usually will not have to pay for that service the working out a tender is. It is a free good from the perspective of the potential buyer. So we cannot construct a traditional demand curve where the demanded quantity of tenders is a function of a price per tender.

What we can do is to consider *the tender and sales department of the company as a profit centre*, a company inside the company which demands tenders because in this way they can get more orders and income which we presume is good for the department and on the other side spend resources to produce tenders. Using standard arguments about marginal income from tenders and marginal costs from producing tenders we can find a quantity of tenders with zero profit.

What can be said about the *demand curve* from the tender and sales department? The lower price the department shall expect to pay for a tender the more tenders it can be expected to demand.

What about the *marginal cost curve*? It can be expected to have the traditional shape as a U – formed curve because from a certain quantity it will be a necessity to pay extra for the competent people to produce more tenders.

3. SOME THEORY

A lot of confusion has often been created in economic analysis from the simple fact that results from micro economic analysis of a single company or a single consumer have been generalized to a whole sector, a whole national economy or the global economy. Also the other way around – from macro economic analysis conclusions have been used on micro level – with the same failures as a consequence.

If we remind us of that warning we can start with *a single company* and ask which preconditions shall be fulfilled for

making it a profitable investment to be done? First of all we shall start with considering our investment plans before decisions have been taken, we are doing an *ex ante* analysis. Here investor can compare expected rate of return from the investment considered with expected rates of return from other investments including the best alternative. The real economic cost is the considered cost plus the difference between return from the best alternative and the considered investment, the so called *opportunity cost.*

If we look at an investment which has taken place, where the building, equipment or product configuration system has been constructed or installed and has been finished, we are talking about an *ex post* analysis. Naturally an *ex ante* preferred investment project can in an *ex post* analysis be a very bad investment. Or an ex ante project which seems bad can *ex post* be a good project. That is a risk or opportunity which is inherent economic life.

Let us now look at the same phenomena on *macro level*. Here we can use the same concepts as mentioned above. But here we can make an observation of some importance. On social or macro level we can only get an improvement in profitability if cost has been diminished. But on a micro level one company can use an investment to establish barriers of entry from competing companies. In this way it can be possible for a company to raise its prices and profitability from an investment even if growth in profitability comes from strengthening of market power and not from higher efficiency in use of the investment. That means what can be an improvement on company level can be a non-plus game on macro level.

Further we can say something more concrete if we look at investment in activities with strong swings in utilisation e.g. in companies producing capital goods or systems which are constructed to produce tenders for capital goods. Here we can find a trade off between two different technologies – one is very labour intensive and

another is very capital intensive. With a given demand for services there will be a certain level where the capital intensive technology will give less unit cost than the labour intensive technology. With a demand less than this equilibrium quantity the labour intensive technology will give less unit cost than the capital intensive technology. With strong swings in demand for our service from capital goods we can ask what to do in respect of technology choice. In principle several solutions can be considered. For example the basic production could be made by the capital intensive technology and the swings could be produced by the labour intensive technology. But in a market economy in contrast to a planned economy there is no organisation to make such a decision or a mechanism to give the same result.

Therefore a more realistic approach to analyse investment decisions during such circumstances is *decision making under uncertainty*. In literature there are two different advices to minimise uncertainties. First of all to *modularise technologies* in order to be able to stop an investment. Second to use *General Purpose Technology* which mean that technologies and modules can be used in several different applications.

4. COMPETITION BETWEEN TRADITIONAL WAY TO GIVE TENDERS AND PCS

If we compare the demand curve and the marginal cost curve in what we have called the *traditional way* and the *PCS way* we can in a stylised way say the following:

<u>The traditional way</u>

- The tender will be very detailed.

- The tender takes the time it takes to get information from many different specialists.

- The marginal costs will be fast rising from a certain quantity of tenders.

<u>The PCS way</u>

- The tender will detailed but not in a one-to-one way with the final detail tender.

- Most of the information about the modules of the good is contained in the model.

- The marginal costs are constant and low for all quantities of tenders.

Reminding that we are looking at *complex products* and assuming that *quality of tenders* are equal we can conclude that we have the traditional situation well known from industrial production that the traditional way for production of tenders is labour intensive with small fixed costs and high marginal costs and the PCS way is capital intensive with high fixed costs and small marginal costs.

However it is important to notice that the PCS tenders compared with the traditional tenders are:

- Much faster from beginning to end

- Less failures in the tender

- Lower marginal costs

An important methodological remark about comparison between the traditional way and the PCS way is the following. If we shall make relevant comparisons we shall compare either the best theoretical cases for both technologies or comparable empirical cases for both technologies. It is scientific fraud to compare average or worst cases from traditional way with best theoretical PCS case. Nevertheless in propaganda this method can often be found.

Another phenomenon to be considered is the fact that technologies with still working investments in fixed capital or human capital, so called sunk costs, are competitive as far as they produce a revenue equal to or more than the variable costs.

If we take a closer look on competition between the traditional well established way to give tenders to potential customers and the PCS way we can describe the competition in the following way:

1. In an investment project where there *ex ante* is no sunk costs or complementary assets investor can choose between investment in the traditional way and investment in the PCS way to give tenders to potential customers.

- The rational decision will in this case depend on the extent and character of *uncertainty* on respect of the two alternatives. Even if our investor is without sunk costs and complementary assets the social dimensions of uncertainties are different because there is much knowledge in other companies and in universities especially for the well established version. So we can say that even if the subjective uncertainty is identical in the two cases the objective uncertainty is less in the traditional way because much knowledge has been accumulated about that from learning processes.

2. In a company with a growth perspective there will ceteris paribus be an argument for the PCS vis-à-vis the traditional way to give a tender because of the differences in fix costs and marginal costs.

5. TWO CASE STUDIES

We shall look at two of our twelve companies described more detailed in section 5. We have selected our cases not because they are representative for all companies but from their differences in showing variety in important dimensions. Our two cases described in the following are examples of engineering intensive companies.

5. 1 Company A

The company has a history of three – fourth of a century starting as producer of components for later to be a supplier of turnkey factories. These turnkey factories are complex in their technology and have been tailored according to wants from customers. The company is today a part of an international group with headquarter in Europe.

The traditional way the company has drawn up tenders for customers has been that a so called calculator as the key person together with the salesman and technologists produces a tender. During several years the work has been done in a systematised partly modularised way where the calculators have had much knowledge often in form as tacit knowledge.

A problem which during the later years has been more and more acute is a problem to find people with interest in the calculator function. Now in the beginning of 2000 years most of the calculators are near to their pension age. So it is of critical importance to preserve the knowledge these calculators have in their minds, to a certain degree as tacit knowledge. That need for the company has been the most important single factor behind the initiative to develop and take into use a PCS.

The company has first developed the PCS during the end of 2003 and the implementation will take place during spring 2004. However there is some interesting information including figures for development costs.

If we take all work done before implementation shall begin the technical support manager assess that about 30,000 working hours or six man year have been spent on understanding what a PCS is and can be used for and after that the more concrete development work. The company representative assess that these 30,000 hours have a value on DKK 12 – 15 mio. Consultance assistance has been used for a value of about DKK 3 mio. Hardware bought is about DKK 500,000.

It shall be mentioned that a benefit already seen outside the direct expected benefits are coming from the process of looking at all parts and structures in the company. Here they have found many forms of slacks.

5. 2 Company B

The company has a history of more than one hundred year in which it has been world leader in its production of equipment often delivered installed in turnkey factories. These turnkey factories are very complex and has until few years ago been designed and constructed as unique in many dimensions. The company is a part of a Danish multinational group.

A problem has been felt by the company during many years consisting in the fact that it took long time and was very complicated and expensive to give an answer about price and time for delivering a turnkey factory. And further in many cases the tender did not result in a contract. The development began in year 1998 when two students from Technical University of Denmark asked about using the company as a case in a master thesis about product configuration. The people in the company with responsibility for producing tenders to potential customers had to use very many resources in fact the same amount of resources as used when a final detailed tender was drawn up. An important dimension of drawing up tenders was the fact that demand for new factories presents extreme swings which naturally can be identified in enquiries about tenders. And the problem was felt in the whole company because many different specialists should give their contributions to make a tender. It was the merging of interest from two students and a growing problem inside the company which was the impetus for taking the decision to allocate resources for construction of a model which could be used to make tenders with.

If we shall understand what happens during transformation from old way to produce a tender to new configuration system exclusive or inclusive what happens outside the tender process in narrow sense we have to look at three dimensions: 1) technology, 2) organisation, and 3) economy. But it shall be stressed that there were and still are two types of tenders before and after introduction of PCS – budget tender and detail tender. In old days before introduction of the first PCS – system the budget tender was identical with the detail tender. After introduction of the PCS a distinction has been introduced between budget tender and detail tender. The difference is first of all that the modules used in budget tender are more ideal than modules used in detail tender. But the difference in price between the two tenders is typically about 5 % and never more than 8 %. Therefore it is acceptable to give budget tender from the PCS. First of all because gains in speed, costs and quality of drawing up tenders.

Experiences from implementation of the first version of a PCS in year 2000 have been in dimensions of relevance for changes in speed, productivity and quality in working out tenders:

Speed in working out tenders:
Before introduction of the PCS:
First tender: eight weeks; people from many departments in company took part in work.
Meeting with the customer: many proposals for revisions.
Second tender: eight weeks; people from nearly all departments in company took part in work.
After introduction of the PCS:

One week and five persons because a de luxe edition of tender was made. A more standardized version would take less than a week and only three – four persons.

Productivity in working out tenders:
Before introduction of PCS:
First tender: 2682 hours from 56 different persons. With a cost per hour on DKK 400: DKK 1.1 million.
Second tender: 1795 hours from 34 different persons. With same cost per hour: DKK 0.7 million.
After introduction of PCS:
44 hours and as a standard edition less. With same cost per hour: DKK 17,600.
Costs for development of the PCS: 200 Person Weeks with same cost per hour: DKK 3.0 million and other costs DKK 1.2 million; total costs: DKK 4.2 million.
There are other costs for maintenance and further development. The estimates given are 10 – 30 person weeks or in money term: between DKK 148,000 and DKK 444,000.
It shall be noticed that in case of customer accept of the tender then there shall be worked out a detail tender which still is more or less the same as before introduction of PCS

Quality in working out tenders:
Before introduction of PCS:
Good and very detailed tender delivered.
After introduction of PCS:
Good to give budget tender, but not so good where customer comes with his tender. Easy to give alternatives to earlier tenders. Customers will not disappear because of time and inflexibilities to deliver tender.

5. 3 Some comments
In both cases we have companies which can best be characterised as engineering intensive producers which mean that they use relatively many engineering hours in tenders, design and even during the construction phase. What we can observe as interesting is that incentives to develop and introduce PCS are perceived as very different in the two cases:
In company A the most important argument mentioned for developing and introducing a PCS is fear of loosing critically *knowledge about drawing up tenders*. In company B the main argument is the possibility to *accelerate speed* in the process of drawing up a tender combined with the possibility of diminishing costs with drawing up a tender.
In both cases we can observe that effects on productivity are not only found in acquisition work but in many other activities in companies. Some of them are in fact found simply because the systematic examination of many activities and their reciprocal connections shows duplication of activities and *slack* of different types. Management will be aware of that and potentialities for improvements in products, processes and coordination technologies and manners.
In continuation of the screening process it will sometimes be evident that integration between different activities sometimes can give very high rates of return. Partly because information and knowledge are so-called *public goods* which mean that they are no – rival in use that means they are free for further use without loss for other part of the company. A consequence from this is that knowledge from the ERP (Enterprise Resource Planning) system can be used to deliver information to the PCS. Contrary need from the PCS on modularity can make it important that design of products is in accordance with some standard.

6. OUR FIRM DATA

12 firms have participated in the study with 30 interviews covering 39 individuals, resulting in more than 45 hours of taped interviews[17]. What follows is a brief description of the participating firms, a description that is made anonymous by request of the participating firms. The firms are grouped after the company type, *before* they introduced product configuration, and the categories are: Heavy engineering, Mass producers, Batch producers, or One of a kind producers. This division is chosen, because it reflects on the production processes and the type of products produced. Heavy engineering firms have no continuous production and essentially make one of a kind, although based on proven concepts. Firms are anonymous and referred to as A, B, C… etc.
Firm A is engaged in heavy engineering producing large production plants, where orders typically ranged from 27 million to 100 million Euros. The main problem in these firms was the cost of producing a tender, which could in the worst case cost up to 4.500 engineering hours, thus putting a significant strain on the organisation. Firms A experienced total costs for developing and implementing their product configuration systems of approximately 1.6 million Euros, and the project lasted about three years. It should be noted that firm A at the time of interview was in the process of implementing a product configuration system. Firm B is also engaged in heavy engineering and experienced problems producing tenders at the rate required by the market. In year 2003 their product configuration system processed quotes for 4,4 billion Euros.

---

[17] The text and data here and in the two following sections build very much on Kasper Edwards and Jesper Riis: Expected and Realized Costs and Benefits when Implementing Product Configuration Systems, 2004.

Firms C, D, E, F, G have traditionally been mass producing and have turnovers ranging from 12,5 million Euros to 600 million Euros and 166 to 3.765 employees. Quite a diverse bunch and all well positioned in their market and some market leaders.

Firms H, I, J are batch producing with one firm (I) sometimes modifying their product to such a degree that one of a kind production might also be a suitable description. Turnover in this group was from 550 million Euro and 801 employees to 22 million Euro and 166 employees.

The firms can give score figures from 1 to 5 where 1 equal "very small" and 5 equal "very large". And 0 equal if the answer is "without influence"
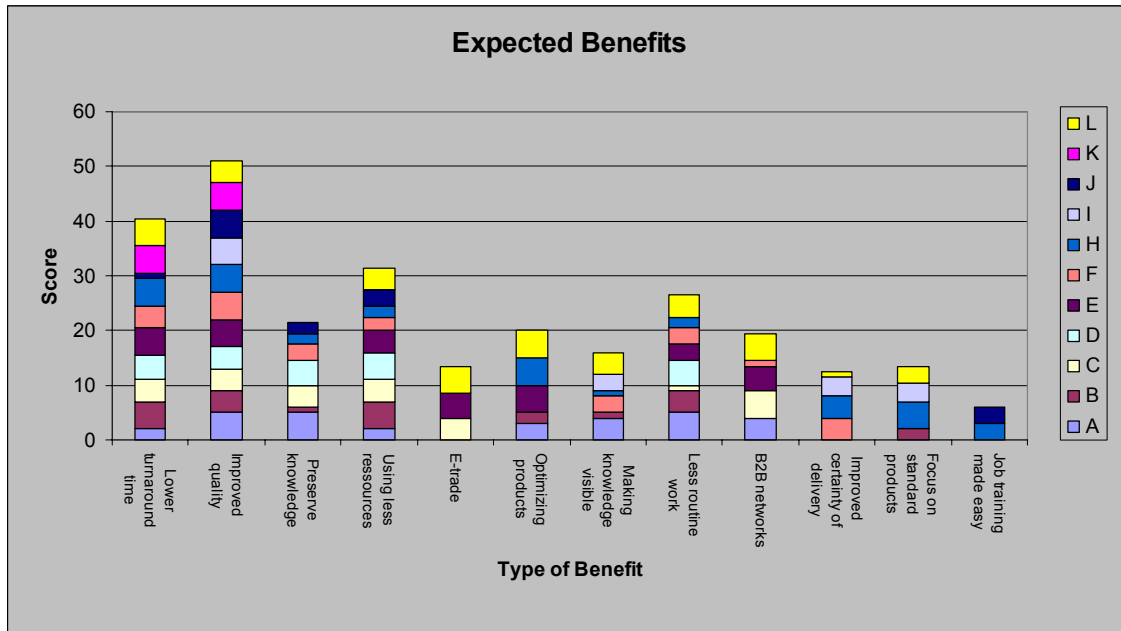

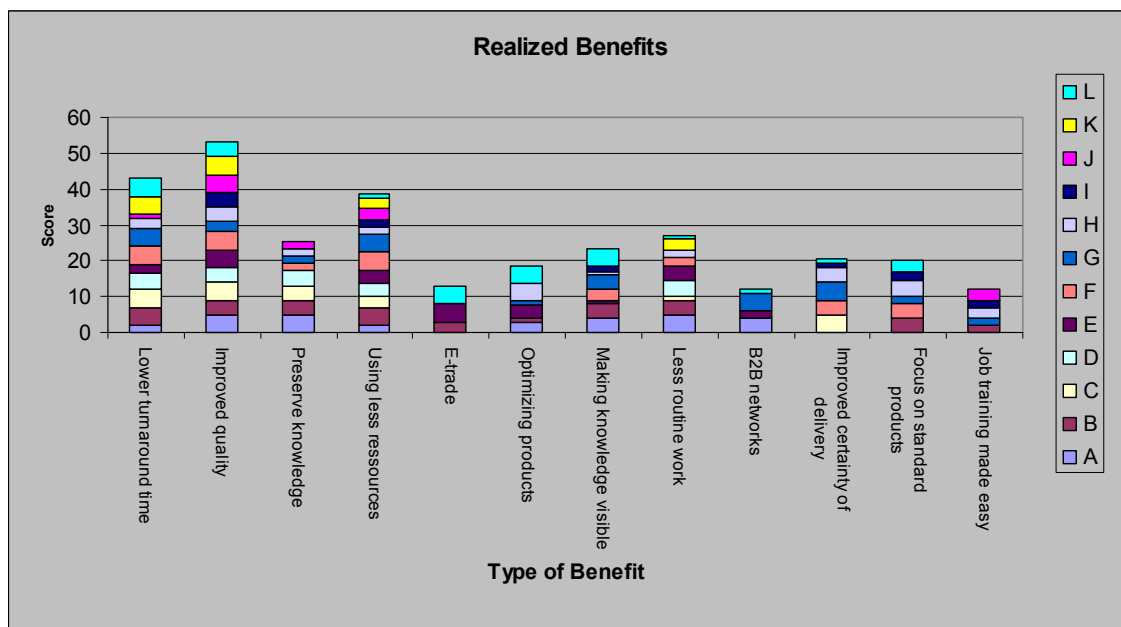**Figure 1: Expected benefits listed in aggregated stacks.**

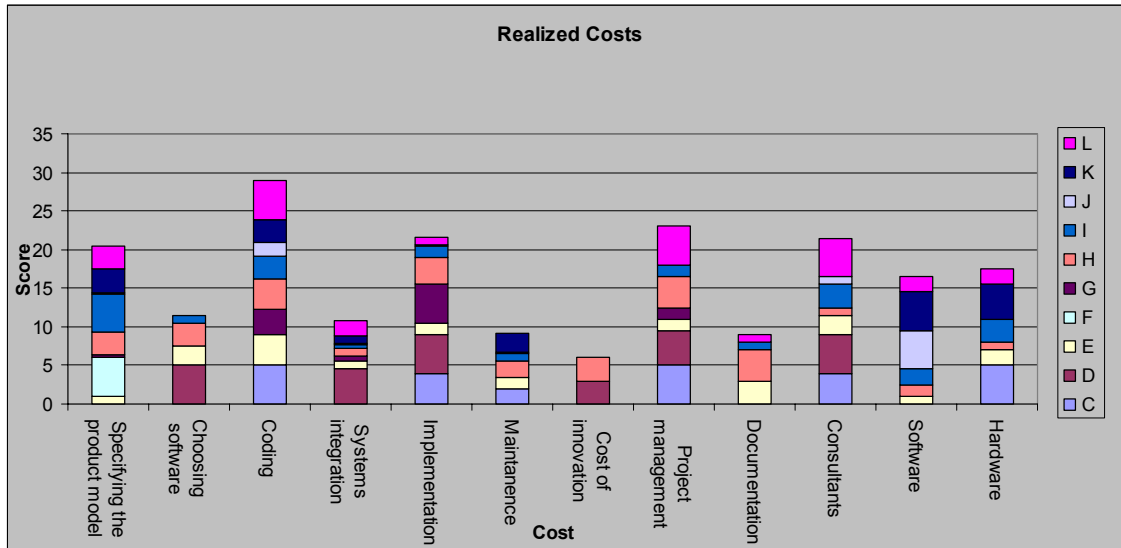**Figure 2: Realized benefits listed in aggregated stacks.**

Figure 3: Realized costs listed in aggregared stacks.

Firm G was the only firm where the listed expected benefits did not match at all. Firm G implemented product configuration with the sole intent of improving inter-company sale, and their second cited reason was to gain complete insight in their production plants across Europe. Firm G had observed that in some cases a customer demands a product which could not be produced in the local company, in which case a sister company in another country was approached by the local company. As the inter-firm profit is lower than regular sales, such requests were frequently delayed to the point, where the customer took his order and placed it elsewhere. The solution was a product configuration system to be use in all sister firms across Europe. The system provides a configuration system which is not tied to the local production system, and a sales person in Austria may configure and allocate production resource in Denmark. This benefit was not observed in other firms and not added to the list of items in the belief that this is a special case.

7. ANALYSES

In this section we analyse firstly the correlation between expected and realized benefits and secondly the correlation between expected and realized costs. Costs and benefits are listed in the same order as shown in figure 1 and figure 2. A ranking of benefits can not be made, as the total aggregated score of benefits provide little insight into the effect of a product configuration system, as one firm might have had expectations and another accidentally realized the benefit. As we are interested in the effects of product configuration systems, it is the correlation between expected and realized costs within the individual firms that will be analysed. The collected data represent the respondents' interpretation of expected and realized costs and benefits in the interview situation. For this reason it makes little sense to use and present a rigid statistical analysis, which would only dilute the reader as to the confidence that one might place in the data. The raw data can be found in appendix.

7. 1 Benefits

12 firms have participated all of which have answered questions about both expected and realized benefits. The benefits are grouped in three categories: High importance representing four or five points, medium representing two and three points and low representing zero or one point. Firms that awarded high importance to a benefit had the feeling that this was an incentive for implementing PCS and important for the success. Medium importance was given to benefits that were reported to be interesting but not critical and not used as a core argument for implementing PCS. Benefits who received low importance were unimportant to the project and were never mentioned as an incentive but none the less nice, if realized.

Firm G holds a special position in this analysis, as driving incentive for implementing PCS was to improve inter-company coordination.

Lower turnaround time was an important expected benefit to 8 of 12 firms, medium important to two firms and unimportant to two firms. All of the firms, which found this benefit important, experienced that customers were lost because of the turnaround time for producing a tender. The two firms I and J that found this unimportant had particular reasons for this. Firm I had just implemented a new ERP system which also focused on reducing

turnaround time. Firm J had a specific problem with the quality of their product specifications, but the time to produce the specifications was not a problem. Only two firms, E and H did have their expectations met.

Improved quality was an important expected benefit to all but one firm (firm G). The importance of improved quality is explained by the consequences of poor product specification quality. Product specifications that are not correct and require correction will be increasingly costly to fix, as the product passes through the production process. In for instance, firm F reported that incorrect product specifications could lead to a complete production halt on night shifts, when key staff (engineers) was off duty. Implementing a production configuration system, in this case, raised the rate of correct specifications from 60 % to 100 %, and no production stops had been reported since. All but two firms achieved their expectations, and the two was just marginally below (one point).

Preserve knowledge was an important expected benefit to three firms (A, C and D) and of medium importance to 3 firms and unimportant to 7 firms. Interestingly enough firms A, C, and D did so for different reasons. Firm A needed to preserve knowledge because of a generation gap in the organisation and foresaw the upcoming pension of a major part of their key engineers. This potential problem had to be countered by using an information system coupled to the configuration system. Firm C needed to allow sales staff to easily access knowledge of the different product variants. Firm D sells high quality, expensive durable goods, which are linked together to form a system, and customers over time may buy additional products and link into the system. Firm D uses the product configuration system to keep track of valid old configurations allowing sales staff to quickly answer questions about integrating a new product into an existing system of older products. As new products offer new and improved features sales staff must precisely be able to identify what features integrate seamlessly and what not thus allowing customers to make an informed decision. All but one firm had their expectations met.

Using less resources was an important expected benefit to 4 firms, of medium importance to 4 firms and unimportant to the last four firms. The firms, which found this important, were experiencing that producing a tender was a significant cost driver and had to be reduced. In particular firms B and L experienced that the tender/order ratio had been declining over that past 15 years, and this was becoming a problem. Using less resource was found to be linked to turnaround time in the sense that if less resource (different staff) were involved, the turnaround time would drop, simply because of the reduction in the number of times a tender would have to wait for staff.

E-trade was an important expected benefit to 3 firms (C, E and L) and unimportant to the remaining 9 firms. Firm C expected to make their configuration system available on the internet but decided later against it and thus did not achieve their expectations at all. Firm E achieved their goal and felt this was important due to their sales organisation. Firm E had an autonomous sales organisation dispersed with offices in many countries over the world which had their own IT systems for configuring and ordering. The configuration system was to bypass some of the local IT systems and allow for a common interface to configuration and also allowing the firm E to control the sales process. Firm L from the beginning envisioned their production configuration system to be available on the internet and hoped to reduce load on sales staff by allowing customers to configure and order the products without intervention by sales staff. All of the other firms did not want to allow customers to access their product configuration system and used it as an internal tool. Firm H based their business on e-trade but had already this capability and therefore rated it as unimportant in relation to their product configuration system.

Optimizing products was an important expected benefit to three firms (E, H and L), 2 firms found it to be of medium importance, and the remaining 7 said it was unimportant. Firms E and L have their configuration systems available on the internet, and therefore it is important that customers can use the configuration system as a means of optimizing their product choice. Firm H makes their product configuration system available to their sales offices, and it is important to use the product configuration system to guide the sales staff to the right product for the particular situation.

Making knowledge visible was an important expected benefit to 2 firms (A and L), 2 firms found this to be of medium importance, and the remaining 8 found it unimportant. Firm A was the heavy engineering firm focused on preserving knowledge. However, preservation was not enough and their knowledge should be easily available to all employees. To leverage this, their product structure was organised on Lotus notes allowing employees to view a particular product, identify parts and their relation to other parts as well as key staff with knowledge about the particular part. Firm L produces a complex product with many rules for its composition, and these rules has been integrated in the product model. Customers and staff should be able to access these rules when configuring a product who's' configuration conflicted with one or more rules and offer a reason and a possible solution. Both had their expectations met.

Less routine work was an important expected benefit to 4 firms (A, B, D, and L), 3 firms found this to be of medium importance, and the remaining 5 found it unimportant. The four firms which found this important experienced a large amount of repetitive work in the process of producing a tender. Firms A, B, and D achieved the expected benefit, but not firm L. While the product configuration system is functional in firm L and is used by customers, the sales staff has not experienced the hoped reduction in routine work. This is mainly because the sales staff does not use the product configuration system and remain working in their old ERP system. Two factors seem to be causing this: 1) A large part of the routine work is processing urgent tenders. A customer calls and asks to have a product delivered within three days, and since there is no spare production capacity, the sales person has to negotiate overtime work and further negotiate a suitable price with the customer.

B2B networks was an important expected benefit to 4 firms (A, C, E, and L), all the other firms found it unimportant. Firms A and E achieved it and allowed other companies to access their configuration system and order products. Firm C abandoned all access to their configuration system from outside agents. Firm L envisioned several companies with access to their configuration system but ended up with a strategic alliance with only one firm. Other firms were in general weary of exposing too much information and knowledge to other firms, be it partners or competitors.

Improved certainty of delivery was an important expected benefit to 3 firms (F, H, and I). Firms F and H reached their goal, and firm I did not. Firm I implemented a new ERP system, which the configuration system was part of. However, the required organisational discipline was not strictly enforced, and staff did not always report status changes to the system, which lead to problems. Although the situation had improved compared to before the system was implemented, there was a clear awareness of the problem of data discipline. Interestingly enough, firm C and G realized this to full effect although not expected. When firm C upgraded their ERP system, the new system contained a new and improved materials planning algorithm, which is the sole reason for achieving this benefit.

Focus on standard products was an important expected benefit to 2 firms (H and I), 2 firms found it of medium importance, and the remaining 8 found it to be unimportant. Both firm H and I realized this, and it had special impact on firm I, which is manufacturing products in which a 40-60 m tower is a central component. It so happens that a tower, which is 2 meters higher that a standard tower may cost more than 100 % more to produce due to changes in the structural dynamics as well as the cost of additional engineering resources to make the required additional calculations. Before implementing the product configuration system this knowledge was not immediately visible to sales staff that accommodated customers to the highest tower. (A higher tower produces a higher benefit to the customer). With the new immediate access to real prices, sales staff can communicate this to customers, who find it difficult to justify a 100 % cost increase in the light of a 4 % increase in production capacity.

Job training made easy was a medium important expected benefit to just two firms (H and J), and the rest of the firms found this unimportant.

7. 2 Costs

The task of analysing cost is somewhat hampered by the fact that some of the responding firms did not report expected costs.

Specifying the product revealed that firms are actually good at predicting the cost of specifying the product. In some cases (firm F and I) this can be time consuming and very expensive. Firm F estimated that 8 person years had gone into specifying their product. For firm I this was also a painful task, as the firm was used to a very fluid understanding of their product, thus documenting that the product and related processes turned out to be very consuming.

Choosing software is interesting, as some of the firms did not realize the importance hereof. Firm E in particular set out to use the Baan configurator but found after a few month of work that integration to their ERP system was difficult, and they decided to use the configuration system integrated in their ERP system.

Coding is high ranking in both realized and expected costs. What is interesting is the systematic misjudgement of the required resources to code. Firm C expected this to be fairly cheap but found that the bulk of costs were in fact related to coding. The lesson here is to be aware of the cost of coding, like software projects' complexity is high, and attention to detail is paramount, which makes it difficult to predict.

Integration to existing systems, on the other hand, is much more straight forward, in particular when using a configuration system, which is part of the company ERP system. Because the interfaces are often specified, it is possible to predict the amount of coding necessary to integrate with existing systems.

Implementation, i.e. the cost of training can also be a surprise to some firms. In particular firms with users in different countries (Firms C, D, G, H) realize high costs. This is related to different organisational setups, different computer systems and to some extent different cultures.

Maintenance costs are in general low. Some firms rated maintenance costs to zero, which is disturbing, as a product configuration system must be maintained to be useful. Follow up questions revealed that the true cost of maintenance was present but too low to be significant compared to other costs.

Cost of innovation was only given a rating by two firms. The remaining firms gave this rating zero points. The two firms recognised that they had to use resources to obtain information about the market, because of the changed specification process. However, most of the responding firms are new to configuration, and this may impact the answer.

Three out of four firms underestimated the cost of **project management**.

The highest expected costs were **Documentation**, which interestingly enough is one of the lowest realized costs. All but one firm expected this to be fairly costly but decided to not document because of pressing schedule and lack of resources. It is expected that firms not documenting their system will have a potential maintenance problem, if key employees leave the firm.

Consultants are, not surprisingly, expensive. Two firms (C and D) miscalculated the need for consultants, although the two situations are very different. Firm C needed to meet a target deadline and did not have the required in-house resources, which lead to use of expensive consultants. Firm D did not specify clearly what jobs

the consultants should do and more importantly not do, in which case the consultants kept working on the system.

Software costs are also very low, which was not expected. This is related to the fact that many of the interviewed firms use the configuration system present in their ERP system. This provides the firms with a configuration system at no additional cost.

The cost of **hardware** was only rated high in one case, and others rated it low compared to other costs. Four firms found the hardware costs to be negligent.

8. DISCUSSIONS

A critical problem in drawing up a PCS is to find information about the often very complex products to modularise these. This information has often been a combination of tacit knowledge in possession of one or very few sellers or has a character of non specified knowledge.

Describing a product in a modularised way will diminish uncertainties connected with drawing up a tender simply because impressions will be substituted by knowledge.

Further the process of making information often in form of tacit knowledge into explicit knowledge will tidy up in components and versions of products without much commercial relevance.

9. CONCLUSIONS

This paper has reported findings on costs and benefits when implementing product configuration systems. Product configuration systems have been characterised, and their relationship to mass customization explained. No less than 12 firms have been interviewed, which provided the data for expected costs and benefits. The data gathered and used is predominantly qualitative and a combination of interviews and quantitative scoring was employed. The quantitative approach to gathering data on costs and benefits is in fact a crude ranking when implementing product configuration systems. The weakness is that 'real' cardinal factual data are not collected. However, it was not possible to actually gain access to these numbers and in most instances firms have not have a quantitative measurement of turnaround time, amount of resources used etc. What firms do have is an understanding of the market in which they sell their products and the firms have a clear understanding of, for instance, if their turnaround time is too high compared to what is acceptable in this particular market. Rankings are well suited for gathering and making a pseudo quantifiable measurement of the costs and benefits from implementing product configuration systems.

It has proven particularly difficult to gather information on the exact or even rough estimates on the expected costs. There are two reasons for this: 1) Some firms have used product configuration systems for more than 15 years and the knowledge is no longer present in the firm, and 2) Exact calculations were not made before initiating the project and the project was based on a rough estimate.

All the firms provided data on the expected and realized benefits from implementing product configuration systems. And from these it is possible to draw come conclusions regarding the primary incentives for implementing product configuration systems. The three highest aggregated scoring expected benefits with more than 30 points from implementing product configuration systems are: 1) Improved quality in specifications, 2) Lower turnaround time, and 3) Using less resources.

It is obvious that the interviewed firms have a problem with the quality of their product specifications and a product configuration system solved this. The product configuration system affects the quality of specifications in two ways: 1) Validate configurations against a set of rules and 2) Ensure all relevant information from the customer is collected. The latter could be achieved with a simple list; however experience has shown this is not enough. Using a configuration system it is impossible to complete a product configuration without all the required information and this forces employee to do so.

Lower turnaround time is a direct consequence of automating the process of configuration a product. Allowing a computer system (product model) to validate a configuration instead of a fellow employee is much more efficient and also does away with a potential queuing problem.

In the same vein using fewer resources is a consequence of automating the process of configuration a product. It is, however, interesting that this benefit was not the primary motivating factor and this is perhaps to be found in the fact all of the interviewed projects engaged the projects from a technical standpoint.

10. REFERENCES

Davis, Stanley, 1987, "Future Perfect"

Dixit, Avinash K. and Robert S. Pindyck, 1994, "Investment under uncertainty",Princeton UniversityPress.

Duray, Rebecca, Peter T. Ward, Glen W. Milligan, William L. Berry, "Approaches to mass customization: configurations and empirical validation", Journal of Operations Management 18 (2000) 605-625.

Empirical evidence from the PETO project (Product Models – Economics, Technology and Organisation). More than 30 interviews are planned and in the process of being conducted in Danish companies. Currently 9 interviews have been completed.

Edwards, Kasper, Lars Hvam, Morten Møldrup, Niels Møller, Jørgen Lindgaard Pedersen, Jesper Riis, "Teoretisk Foranalyse", The PETO project, Department of Manufacturing Engineering and Management, Technical University of Denmark, 2003.

Edwards, Kasper and Jesper Riis, "Expected and Realized Costs and Benefits when Implementing Product Configuration Systems", International Design Conference – Design 2004, Dubrovnik, Croatia 2004.

Eisenhardt, Kathleen M., 1989, "Building Theories from Case Study Research", Academy of management Review, Vol 14, No.4, pp.532-550.

Faltings, B. and Rainer Weigel, 1994, "Constraint-based knowledge representation for configuration systems" Technical report No. TR-94/59, Department D'Informatique.

Forza, Ciprane and Fabrizio Salvadore,2002, "Product configuration and inter-firm co-ordination: an innovative solution from a small manufacturing enterprise", Computers In Industry 49 pp. 37-46.

Forza, Ciprane and Fabrizio Salvadore, 2001, "Manageging for variety in the order acquition and fulfilment process: The contribution of product configuration systems", Int. J. of production economics 76, pp. 87-98.

Hansen, Benjamin Loer, 2003, "Development of Industrial Variant Specification Systems, Ph.D. thesis, Department of Manufacturing Engineering and Management, Technical University of Denmark.

Hvam, Lars, 1994, "Anvendelse af Produktmodelering – set ud fra en arbejdsforberedelsessynsvinel", Ph.D. Dissertation, Driftteknisk Inistitut, Technical University of Denmark.

Hvam, Lars, 1999, "A Procedure for building Product models", Robotics and Integrated Manufacturing.

Jacobsen, Jan K., 1997, "Interview – Kunsten at lytte og spørge", Hans Reitzels Forlag (in Danish)

Kvale, Steinar, 1997, "InterView", Hans Reitzels Forlag (in Danish) translated from English: Nake, Bjørn, 1994, "InterView"

Pine,B. Joseph II, 1999, "Mass customization – The New Frontier in Business Competition", paperback, Harward Business School Press.

Riis, J. 2003,(In Danish) "Fremgangsmåde for Opbygning, implementering og vedligeholdelse af produktmodeller", Ph.D. Thesis, Department of Manufacturing Engineering and Management, Technical University of Denmark.

Silveira, Giovani Da, Dennis Borenstein and Flávio S. Foglitto, 2000, "Mass customization: Litterature review and research directions", Int. J. of production economics.

Schwartze,. S, 1996, "Configuration of Multiple-variant products", BWI, Zürich.

APPENDIX

Expected Benefits	A	B	C	D	E	F	G	H	I	J	K	L
Lower turnaround time	2	5	4	5	5	4	0	5	0	1	5	5
Improved quality	5	4	4	4	5	5	0	5	5	5	5	4
Preserve knowledge	5	1	4	5	0	3	0	2	0	2	0	0
Using less resources	2	5	4	5	4	3	0	2	0	3	0	4
E-trade	0	0	4	0	5	0	0	0	0	0	0	5
Optimizing products	3	2	0	0	5	0	0	5	0	0	0	5
Making knowledge visible	4	1	0	0	0	3	0	1	3	0	0	4
Less routine work	5	4	1	5	3	3	0	2	0	0	0	4
B2B networks	4	0	5	0	5	1	0	0	0	0	0	5
Improved certainty of delivery	0	0	0	0	0	4	0	4	4	0	0	1
Focus on standard products	0	2	0	0	0	0	0	5	4	0	0	3
Job training made easy	0	0	0	0	0	0	0	3	0	3	0	0

Realized Benefits	A	B	C	D	E	F	G	H	I	J	K	L
Lower turnaround time	2	5	5	5	3	5	5	3	0	1	5	5

Improved quality	5	4	5	4	5	5	3	4	4	5	5	4
Preserve knowledge	5	4	4	5	0	2	2	2	0	2	0	0
Using less resources	2	5	3	4	4	5	5	2	2	3	3	1
E-trade	0	3	0	0	5	0	0	0	0	0	0	5
Optimizing products	3	1	0	0	4	0	2	4,5	0	0	0	5
Making knowledge visible	4	4	0	0	1	3	4	1	2	0	0	5
Less routine work	5	4	1	5	4	3	0	2	0	0	3	1
B2B networks	4	0	0	0	2	0	5	0	0	0	0	1
Improved certainty of delivery	0	0	5	0	0	4	5	4	2	0	0	1
Focus on standard products	0	4	0	0	0	4	2	4,5	3	0	0	3
Job training made easy	0	2	0	0	0	0	2	3	2	3	0	0

Product configuration from the customer's perspective:A comparison of configuration systems in the apparel industry

Timm Rogoll and Frank Piller

TUM Business School, Institute for Information, Organization, and Management (IOM)
Technische Universitaet Muenchen,Leopoldstraße 139, 80804 Munich, Germany
rogoll@wi.tum.de, piller@wi.tum.de

Abstract

This paper aims to transfer some findings from research on configuration systems into practice. It provides a comparison of twelve real examples of (competing) configuration systems for mass customization in the apparel industry from the perspective of the user (customer). This comparison shall help, first of all, managers to get an indication on promising, but also worst practices of online product configuration.

Beneath all technical and functional aspects, a product configuration system must fulfill, first of all, also the demands of the customer (or user) of the system. This task is of foremost importance especially if the configuration system should be used by end consumers directly, e.g. via an internet interface. The objective of our paper is to look on product configuration through the customers' eyes and present a structural approach of success factors of customer orientated product configuration. We will base our research on an exploratory comparison of product configuration systems for custom made shirts. This is a rather developed industry, and a larger number of applications can be found both online and offline. However, from comparing these vendors, it is astonishing that even obvious factors of configuration usability are often not taken into account. Thereby, the most important factor of success is founded in exactly these demands of the customer. We will show on the one hand which requirements are important from a theoretical point of view, and on the other hand how the vendors of customer made shirts do or do not archive these requirements in their configuration systems.

Customer requirements towards a configuration system can be clustered in different categories. These generic categories contain a number of different approaches and guidelines to describe design rules of customer orientated product configuration systems. They are responsible to assure the success of a configuration system. However, as our study indicates, a large problem in this context of customer requirements is that neglecting only one small factor can disturb the whole configuration experience. A system has not to be perfect in only one of the categories. The customer in the environment of the Internet is very sensitive and every even small obstacle can cause the termination of the configuration process − and, thus, the selling process.

Customer orientation in product configuration, thus, is of foremost importance for the growing market of self configurable products and service on the internet and in offline interaction systems. From our experience, the imperfect customer orientated design of many product configuration systems is a major reason why business concepts like mass customization or made-to-measure clothing are behind their market forecasts and expectations of the industry.

Keywords: Configuration, Customer Requirements, Usability, Visualization, Trust Building , Reduction of Risk

This paper describes "research in progress". It is a working paper directed to give practitioners an indication on the design of the frontend of a configuration systems for mass customization in consumer markets on the internet.

Product configuration systems as a main enabler of mass customization

The objective of mass customization is to deliver goods and services, which meet individual customer's needs with near mass production efficiency ([1]; [2]; [3]; [4]; [5]). This preposition means that individualized or personalized goods can be provided without the high cost surpluses (and, thus, price premiums) usually connected with (craft) customization. Therefore, mass customization is often connected closely with the capabilities offered by new manufacturing technologies (CIM, flexible manufacturing systems) reducing the trade-off between variety and productivity and hence enabling to decrease the additional production costs ([1]; [6]; [7]; [8]; [9]).

However, new flexible manufacturing systems are a necessary, but not sufficient condition for successful mass customization. They have to be supplemented by infrastructures capable of handling the information flows and transaction costs connected with mass customization, which is characterized by a high intensity of information compared to mass production ([10]; [11]; [12]). This information intensity results from the fact that starting point of each customization are specific customer needs. The supplier has to interact with each single customer and obtain specific information in order to define and translate the customer's needs and desires into a concrete product specification. Only within the last years sufficient technologies exist to handle the information flows connected with mass customization. This explains the time lag between the first discussions of the concept in literature for more than a decade (e.g. [13]; [14]; [1]; already [15] described the basic idea), and its practical implementation, which took place only in the last few years. Especially as mass customization enters more and more consumer markets, new information technologies can be seen as its main enabler ([16]).

By translating individual customer needs in a specific product, the customer is involved into the value creation of the supplier. The customer becomes a "co-producer" respectively "prosumer" ([15]). However, as the main part of the interaction with the customer takes place during the configuration and therefore the design of a customer specific product, it seems appropriate to call the customer rather a co-designer than a co-producer. Customer co-design describes a process that allows customers to express their product requirements and carry out product realization processes by mapping the requirements into the physical domain of the product ([17], [18], [19]). A product is co-designed between each single customer and the supplier.

Against this background, the importance of an interaction and configuration platform that enables users to design the desired product seems obvious. Every transaction in a mass customization system implies information and coordination about the customer specific product design and is based on a direct communication between the customer and the supplier ([20]; [21]). Thus, the main distinctive principle of mass customization is a mechanism for interacting with the customer and obtaining specific information in order to define and translate the customer's needs and desires into a concrete product or service specification ([21]). Companies offering mass customized products have to develop and operate new kinds of customer interfaces and customer interaction systems building an efficient platform for this value co-creation.

The interaction with the customer is the core business of a mass customization process. Only by matching the knowledge about the capabilities of the vendor with the knowledge about the whishes and demands of the customer, a reasonable fulfillment process becomes possible. The customer's integration in the value creation process of mass customization creates a collaboration between the supplier and the customer, which supersedes the traditional value chain. Companies successfully pursuing mass customization build an integrated knowledge flow – that not only covers one transaction

but uses information gathered during the fulfillment of a customer-specific order to improve the knowledge base of the whole company ([23]; [21]; [4]). During the whole process the interface between manufacturer and customer is crucial. Not only does it comprise the solution space of the production facilities, but it is also the design instrument for new and existing customers and the core communication tool. (e.g. [24]; [25]; [26]).

Known as configurators, co-design-toolkits, choice boards, design systems, platforms, or co-design-platforms, these systems are responsible for guiding the user through the configuration process. Different variations are represented, visualized, assessed and priced, which starts a learning-by-doing process for the user. While the term "configurator" or "configuration system" is quoted rather often in literature, it is used for the most part in a technical sense addressing a software tool.

The success of such an interaction system is, however, by no means not only defined by its technological capabilities, but also by its integration into the whole sale environment, its ability to allow learning by doing, to provide experience and process satisfaction, and its integration into the brand concept. In this environment, main fields of interest are process patterns of user interaction, user satisfaction, perceived risk as well as value of individualization. Only when these basic issues of consumer configuration behavior are understood, appropriate technical solutions and IT-systems for configuration systems can be created properly ([22]).

Many of the prominent examples of mass customization rely on consumer direct, online sales strategies. Consumer goods companies like P&G or Nike, which normally use multi-level retail channels, sell their mass customized products via internet in direct interaction with their consumers. When Nike started mass customizing with NikeID, it decided to offer this product only via its own website. Apart from lower transaction costs due to the "design it yourself" approach, the main motivation behind this decision was to gain experience in the interaction with consumers on the web. Theory also supports this approach. For individualized goods and services transaction cost theory recommends, at first glance, direct interaction between manufacturer and buyer. Configuration and purchasing should be fulfilled without any intermediaries ([30]; [31]). An intermediary would do nothing more than transfer the product specification to the manufacturer where each order has to be checked, planned and carried out separately. Thus, a retail channel or an own offline sales structure would just add an additional (transaction) cost-generating level, especially as today electronic commerce allows manufacturers to communicate and trade with large groups of consumers directly and efficiently. The objective is to keep transaction costs low, while still delivering an intensive interaction and communication with the customer. This is only possible if the interaction is skillfully automated, i.e. becoming a do-it-yourself or self-service solution.

Despite a huge number of variation, the electronic systems within a mass customization interaction platform consist of three main components ([28]; [29]; [4]):

- The *core configuration software* presents the possible variations, and guides the user through the configuration process, asking questions or providing design options. Consistency and manufacturability are also checked at this stage.

- A *feedback tool* is responsible for presenting the configuration. Feedback information for a design variant can be given as *visualization* and in other forms (e.g. price information, functionality test etc.) and is the basis for the trial-and-error learning of the user.

- *Analyzing tools* finally translate a customer specific order into lists of material, construction plans, and work schedules. They further transmit the configuration to manufacturing or other departments.

The focus of our paper are user interactions with online interaction systems, so called (online) configurators or toolkits for user co-design. Our paper will be structured as follows: After this introduction, we will illustrate the importance of interaction platforms for mass customization. In Chapter 3, we will discuss the requirements such a system must fulfill to deal with the specific demands of a mass customization selling process. Next, we will extend this discussion by a taking the perspective of the customer. To illustrate and apply the framework generated there, we will present a comparison of different online configuration systems in the apparel industry for the manufacturing of customized men's shirts.

This paper aims to transfer some findings from research on configuration systems into practice. It provides a comparison of twelve real examples of (competing) configuration systems for mass customization in the apparel industry from the perspective of the user (customer). This comparison shall help, first of all, managers to get an indication on promising, but also worst practices of online product configuration.

General tasks of a configuration systems

Before we discuss the specific requirements towards a configuration system from the perspective of the customer, we will describe briefly some more general tasks a configuration system must fulfill in a mass customization environment. These functions are the basic enablers of an online configuration activity:

1. Presentation of the company and the abilities

2. Presentation of the offering

3. Consultancy and support

4. Guidance in the configuration process

5. Intermediation of a „flow experience"

6. Plausibility check of the selection

7. Privacy of personal data

Presentation of the company and the abilities

Configuration systems, which are used directly by customers themselves, represent directly the company and its capabilities. The website and the integrated configuration system are becoming the

universal communication interface. This, it is important for a mass customizer to design an interface, which gives the customer an adequate and pleasing look on the products and capabilities of the company. As our practical analysis of configurators in the apparel industry will demonstrate later, this very basic requirement is still lacking in many examples. A number of companies place a strong emphasis on the technical product selection process, but lack to demonstrate their capabilities to meet a unique customer request.

Presentation of the offering

A configuration system must be able to show the spectrum of the variation possibilities. The aim is to show the customer, like in the case of a conventional paper-based catalogue, possible specification of the product architecture. Customers have to be educated about the possible configuration possibilities, and have to get an overview about what is possible at a given point of time. It is important to integrate the configuration system at this stage with the ERP system to get real-time data on available components and options. However, beyond presenting the actual features of the modular product structure, the system has also demonstrate the combination possibilities of different options, has to show constraints and limitations, and has to present this information in a way that a rather suggestive configuration process becomes possible.

Consultancy and support

An other task is to consult and support the customer during the configuration activity. Especially with the offering of individualized products the consultancy and mediation of product knowledge plays an important role. The customer is during the whole configuration process in the positions of making decisions he or she perhaps never thought about before, as these decisions were made by the professional designer of the manufacturer. Many configuration steps are also accompanied with possible insecurities. These insecurities often lead to break up the configuration activity if the perceived complexity of the customer is to high. Thus, configuration systems should incorporate not only a mechanism to make selections, but also a solution to assist customers to get ti know their own demands.

Guidance in the configuration process

Based on the complexity of the product architecture, it sometimes becomes difficult to do a complete and correct configuration, even if the customer's knowledge about the product is rather high. A configuration system must be able to guide its users through the whole configuration process. In practice, there are today only very few configuration systems, which deal with the customer's wishes in the very beginning. Most systems confront their users directly with the assortment of different modules or components. Thereby the assistance should be primarily oriented on the usage and requirements of the customer and not on the product structure and architecture.

Intermediation of a „flow experience"

To offer support and guidance during the configuration process is also an important requirement to engage customers into a special purchasing experience. Studies have shown, that the perceived customer contentedness with the product bought in a mass customization process is in very close conjunction to the contentedness adapted during the configuration and selling process itself ([32]; [34]). For the majority of customers, it is a very special experience to participate in the co-design of an individualized product. This fact strengthens the identification and the involvement of the customer towards the final product. A configuration system should be able to create and mediate this enthusiasm. In line with configuration systems, the theory of the flow experience plays an important role ([32]; [34]). The activities along the configuration process should be as experienced by the user as a benefit per se, the process itself shall be experienced as worth doing – and not only as a necessity to get a customized product.

Plausibility check of the selection

During the configuration process, every selection of on option on one stage of the product structure often indicates a limitation, modification, or an alternative for the final product, as based in the product architecture. Selections of options are in many cases not independent, but depended from the selection of other variants at one configuration option. These dependences are incorporated in the product family or product structure. Configuration rules can be implemented according to three different ways [34]:

- *Procedural:* In this way, the configuration process follows a strict hierarchical, step-by-step procedure. Every step calls for the next, and it is impossible to miss out a single step till the end of the configuration process. These are the simplest and easiest configuration systems from the technical point of view.

- *Decision-rule-based systems:* These systems give the user more freedom in the configuration process. In decision-rule-based systems, the configuration rules are put together in form of a network. This allows the user to jump to different configuration steps at any time. This way even allows to change the first selection, even if other configuration steps have already been finished. Most of the online configuration system implemented today can be seen in this group.

- *Knowledge-based systems:* These systems do not present options for the different configuration steps, but inquire in a rather conversational mode the demands and wishes of a customer ant transfer them into a product specification. Knowledge-based systems often have a very complex structure, because they must be able to evaluate and choose components based on the demands and whishes the user has given. A lot of implied and explicit knowledge has to be transferred and translated into rules and dependencies to be used in the system.

Privacy of personal data

The collection of personal data is necessary in every online shop, and a rather sensitive and delicate part which was discussed in extend in the literature before. In a mass customization environment, privacy, however, can become an even more sensitive issue. To produce an individual product like clothes, the manufacturer needs personal data like body measurements. This is information that many consumers want to be know in safe hands as much as their credit card number etc.

The state of implementation in practice

The table below present a number of configuration systems on the internet which incorporate one or several of the steps presented before in a rather good way. However, not one company do perform all of these tasks excellent, often excellence in one area is counterbalanced by weaknesses in another area. Nevertheless, these systems named below provide a good overview ob the state of configuration systems in consumer markets in several industries.

Area of Configuration Toolkit	Examples
Presentation of firm / capabilities	• Procter&Gamble's Reflect.com (www.reflect.com) • BMW Mini Car (www.mini.de) • ic! Berlin Sunglasses (www.ic-berlin.de)
Presentation of offering / presentation of customization possibilities	• CMAX Fashion Shoes (www.cmax.com) • Elkamet Street lightening (www.elkamet.de) • Watches by factory 1-2-1 (www.factory121.com) • Customized wallpaper (www.berlintapete.de) • M&M Colorworks (www.colorworks.com)
Selection of basic model / pre-configuration	• Welcome 24/7 kitchens (www.officekitchen.de) • Modula Computer (www.systemkonfigurator.de) • Officeshop office furniture (www.officeshop.de)
Guidance, Consultancy	• customized apparel Swisstex (www.swisstex.net) • mi-adidas sport shoes (www.miadidas.de) • kitchens (www.kuechen.de)
Leading customer through configuration process	• BMW Car Configurator (www.bmw.de) • Prosche Car Configurator (www.porsche.de) • Dolzer Shirt Shop (www.dolzershop.de) • Watches by factory 1-2-1 (www.factory121.com) • M&M Colorworks (www.colorworks.com)
Auto configuration of missing options	• ABC Computer (http://pc-konfigurator.suk.de) • Dolzer Shirt Shop (www.dolzershop.de) • Prosche Car Configurator (www.porsche.de) • USM Furniture (via www.perspectix.com)
Visualization	• Orako cell phone covers (www.designyourhandy.de) • CMAX fashion shoes (www.cmax.com) • Campe-Ohff shirts (www.campe-ohff.de)
Gain of customer profile	• Dolzer shirt Shop (www.dolzershop.de) • Dell Computer (www.dell.com)
Gain of aggregated customer knowledge	• Procter&Gamble's Reflect.com (www.reflect.com) • CMAX fashion shoes (www.cmax.com) • Selve shies (www.selve.net)

Table 7: Promising practice examples for configuration systems

Customer requirements towards configuration systems

In the last chapter, we have presented a number of task a configuration systems has to fulfill and incorporate in the overall application. In the following chapter, we want to take an ever deeper perspective of the customer. Configuration systems have to fulfill three major tasks from the customer perspective – beyond the sheer ability to create an individual product specification.

Risk reduction and building of trust

From the customer's perspective, the co-design process of mass customization is connected with additional costs ([10]; [23]). Users often have no clear knowledge which solution might correspond with their needs, sometimes they even still have to find out what their needs are. As a result, customers may experience uncertainty during the design process. Uncertainty about the behavior of the supplier exists, too. The newer and more complex the individualization possibilities are, the more information gaps increase. These processes are characterized by an asymmetrical distribution of information – a typical principal-agent-constellation ([27]): Customers (principals) order from a supplier (agent) – and often pay in advance – a product they can only evaluate in a virtual form. Then, they have to wait days or even weeks to receive it. The customers are not in the position to evaluate the product they are willing to buy.

All these facts are aligned with uncertainties which can be interpreted as additional transaction costs for the customer. One major tasks of a vendor is to reduce these costs to a minimum. The benefit the customer adepts from the individualization must be higher than the perceived costs and troubles originating from the customer's troubles in the configuration process. Vendors that offer their customers the biggest possible variety and at the same time give them the best possible encouragement in the configuration process, achieve a competitive advantage. The configuration system must be able to reduce the risk, build up confidence and show competence. One of the most important tasks of the supplier is to ensure that the customers' expenditure is kept as low as possible, while the benefits they experience are clearly perceptible. Configuration systems are the prime instrument for reducing the user's costs arising from a principal-agent-constellation that is inevitable in mass customization. If the customer has trust in the process, as well as in the product he has chosen and his bad feelings about the uncertainties are low, the probability of a finished purchase is rather high.

Usability

Looking on the configuration process itself, usability is defiantly the most important requirement a customer has on a configuration system. Usability describes the relationship between the application and the user. The quality of an application points at the best possible encouragement in the tasks that have to be attended. Therefore usability plays an important role on different levels: "Usability is the

quality of a system that makes it easy to learn, easy to use, easy to remember, error tolerant, and subjectively pleasing" [(36)]. Usability is directly responsible for the success or failure of a configuration system. Because in the perspective of the user usability makes the difference if the process can be performed correct and complete or will be aborted because of too much complexity or obscurity. The best technologies innovative functionalities are useless if the user is not able to deal with them.

In regard to accessing the usability of a configuration systems for mass customization, the following aspects are of predominant importance:

- Operability and Self Explanation

  The intuitive handling and navigation put together with the feeling of success by coming through the configuration process shapes the impression on a configuration system. Studies have shown [(36)] that it is necessary for a good usability that standards, like the color of links or the usage of symbols, already learned by the users help them to navigate and find their way. Operability and self explanation relays at last on a certain degree of guidance. It is important that, the single configuration steps are built up logical and that it is obvious how to navigate trough the configuration process.

- Orientation

  Websites are not physical tangible and exactly in this, a basic problem is located. We need possibilities and methods to show the user or customer a picture of the whole proposal and functionalities the system offers. Only if this picture is given, the user can put is position in relation and is able to orientate and behave. Transparency and traceability of the application structure are basic premises for the orientation and this is necessary to act fast, provident and goal-oriented.

- Individual Access on Information

  A lot of problems with the online interfaces are based on a lack of knowledge about the usage behavior of the target group. The Usage behavior is a complex topic. Each individual process information in different way, accumulates in a different kind and gets back to saved information in his own method. To deal with this multiple treatment of information it is necessary to make the information accessible in any possible access method ([37]). This means Information must be available in different kind of principles like:

  - textual
  - visual
  - alphabetical
  - numerical
  - chronological
  - geographical
  - etc.

- Loading Time

The loading time doesn't play a too important role, particularly because more and more users and customers have a fast internet access. But it is nevertheless in some importance, that the application doesn't take too much time for loading ore actualization after a configuration step. Still most of the breakups on Internet sites are based on to long loading times especially in the combination with plug-ins.

For configuration systems the loading time particular plays a important role incoherence of the visualization, which is one of the major parts in the following chapter (4.3).

- Support

Even that this point is a matter of course and nearly every system on the market has some kind of support functionality, it is unbelievable how useless and problem-unoriented most of these functionalities are. Naturally it is the requirement of the customer to deal with the system without needing any help. This is because of the complexity, of some products and therefore the configuration system as well, impossible. But Support in this case must be able to deal with probable problems and not just giving extra information.

Visualization

In the virtual environment of the internet, it is impossible for customers to build up a relationship to the physical product they are planning to buy. In contrast to the conventional selling process in shops and stores, where the customer is directly and physically face to face with the product or at least a similar product, the judgment in the virtual environment is a very difficult task. A lot of the characteristics, as well as the quality are not feasible, until the final product has been delivered. During this waiting time, characteristics can slip in the mind of the customer and build up a gap between the image of the customer and the real product.

To counterbalance this problem, a virtual product image has to be produced. Visualization as part of the configuration process is necessary to provide the customer a real feeling about the product, to give him a substitute for the not yet existing product. The major point is to illustrate the configured achievement. "What you see is what you get" is the golden rule to give the customer a good feeling about the product, to make him buy and show him competence.

Visualization can be offered at different points during the configuration process. A rather common solution is to present the visualization just at the end of the configuration process, once the process has been finished. Better, however, is to present the user a continuous visual representation of the resulting product during the configuration process, guiding him through every single configuration step by an on-point visual representation of the recent state of specification. Visualization is one of the strongest instruments to create trust and reduce the risk perceived of the user – and to increase the willingness to purchase. There are different ways of how visualization can be technological implemented:

- *Pre-Produced Picture and Drafts:* The presentation of pre produced pictures is a common and of course a good possibility to show the product. The pictures normally taken by photograph and then just linked to the configuration solutions. But also simple drafts are seen quite often. The big problem with this kind of visualization is that each variation has to be produced in first. For

offerings with a large number of Variations this becomes very soon a Problem and causes a lot of cost.

- *Compound Pictures:* In this kind of Visualization the pictures of the product a put together out of single components. Just like a box of bricks the product picture is set together out of the different pictures of components. This has the advantage that not all possible configurations have to be produced and only the visible parts must be indicated.

- *Rendered Pictures:* Rendered pictures is a more advanced way of producing a picture of a product and is some distance to the way of taking photos. Rendered pictures need vector graphics of the product or the components. But these vector graphics don't give a real product view. The surfaces first have to be coated by a texture and then are illuminated by the computer. A lot of advantages and possibilities are with technique convertible. Different views, colors, textures, illuminations can be shown without existing in the real world at all.

- *3-Dimentional Models and Augmented Reality:* The most advanced technique of Visualization is building of 3-dimetional models. Although the structure is close to that of the rendered pictures the effort is much higher. The 3-dimetinal models can normally be free rotated and sometimes even has some of the products functionality integrated. The computing capacity and the complexity of the programs you need to generate such models are very high and expensive. But it gives the customer the closest visual feeling of the real product. Nevertheless the next evolution step in Visualization is already on the way. Augmented reality is the one way, not only to give the customer a 3-dimetional idea of the product, but putting the product in the context of the customers surrounding (for example: Showing the Customer his new kitchen is his flat without building up a single part). In the apparel industry are new ideas, like virtual models which look like the customer and can put on the clothes, he just has chosen and individualized.

Comparison of configuration systems in the apparel industry

In this chapter, we take a closer look at configurations systems in the apparel industry in order to evaluate the state of implementation of theses systems in the industry practice. Our focus are various configuration systems for customized shirts on the internet. We have chosen this industry as here the development has reached a rather mature state. It was one of the first industries going online in the mass customization domain.

We identified the vendors for our evaluation by checking first special portal sites listing mass customization offerings (like digichoice.com or egotrend.de). Secondly, we looked in "Google" for further vendors using the key words (or combination thereof) "shirt, custom, tailored, tailor-made" in different languages. We found a large number of vendors, and most of them exist for several years. In the following, we take a closer look on twelve shirt manufacturers, most of them located in the US and Europe (see Table 2). We selected these twelve systems after a first visit and exploration of the sites. We also selected only systems which can be operated in a language readable by the research team. The evaluation of the sites, which is presented in the following, is based on an expert evaluation of the research team.

Name	Description	Country	Internet
Ravis International	Thai tailor offering custom made cloth (men and women)	TH	www.ravistailor.com/
AmericanFit	women's and men's clothing such as pants, shorts, and skirts. Features a virtual fitting room.	USA	www.americanfitclothing.com
MyTailor	made-to-measure shirts for men, with al big offering of different fabrics	USA	www.mytailor.com
Bivolino	Belgum shirt manufacturer with a great offering and a lot of modern fabrics and styles.	BE	www.bivolino.com
Liste Rouge	A French customizer for men shirts, with strong ambitions for the American market	FR/USA	www.listerouge-paris.com
Dolzer	A German custom-made cloth maker for men and women with a very good online shirt shop	DE	www.dolzershop.de
Dietrich	Shirt manufacturer for men's shirts only, German internet MC pioneer since 1996	DE	www.dietrich.com
Perleth	Shirt manufacturer for classic men's shirts (Frankfurt)	DE	www.online-shop-masshemden.com
Gert Keller	German manufacturer for men shirts from Berlin	DE	www.gertkeller.de/
Holl	Italian shirt manufacturer especially for men shirts an accessories (Boxershorts)	IT	www.holl.it
Stefanelli	Traditional shirt manufacturer from Firenze	IT	www.stefanelli.it
The Shirt Tailor	A very compact configuration tool for men shirts	USA	www.shirttailor.com

Table 8: Configuration systems covered in the comparison

In the following, we will take the demands on a good online configuration site described in Chapter 3 above and show both a promising practice and a rather weak practice from our field. The idea is to give practitioners an indication on how to fulfill the consumer demands towards configuration systems for mass customization

Examples for Operability and Self Explanation

The following picture present two examples for operability and self expiation on the same website. The left picture shows how easy and unmistakable a configuration step can be designed. The only

functionality on this page is the selection of a material. The selection takes place just by clicking on the preferred example. There is no need to explain this step, and the user is automatically guided to the next step after the election.

However, while the vendor managed to design this selection step easy and self-explaining, it is part of a larger process which is much more difficult to navigate. On the right side, one can see the previous configuration page. Al lot of categories and notations, none of them give any idea of what hides behind these. A lot of links, making no sense at all for an inexperienced users. Further, there is no information available about these links at all. For the user who is in the search for example a yellow shirt, this site is not able to assist.
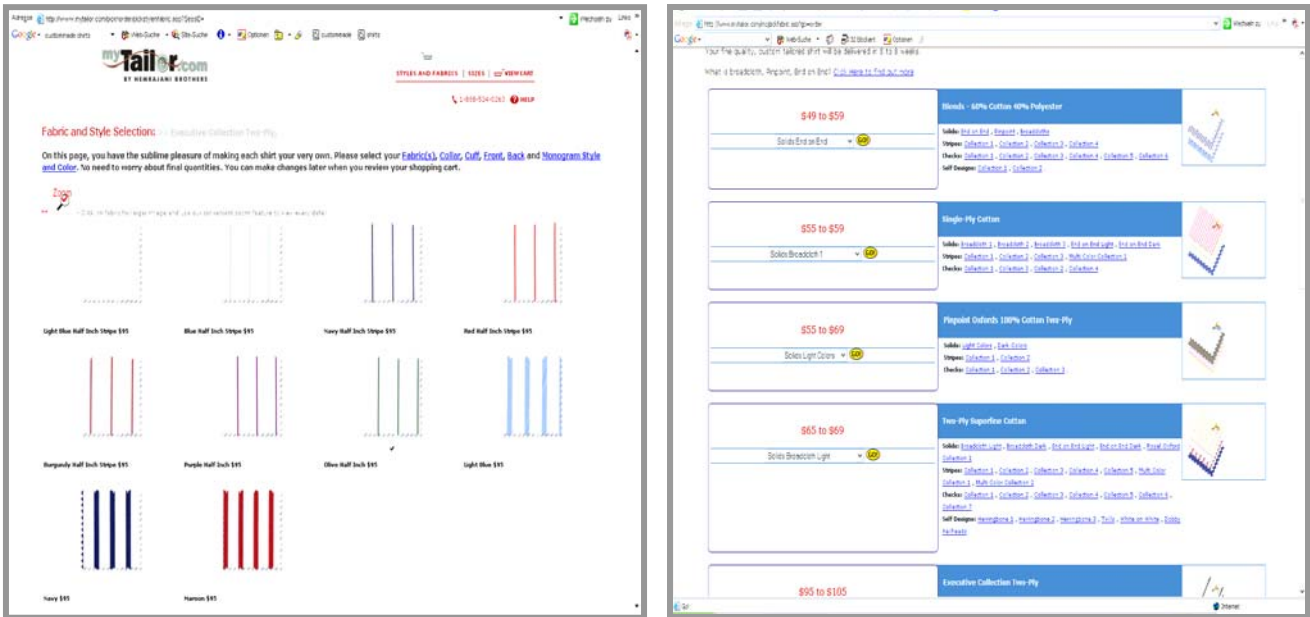
Picture 1: Examples for Operability and Self Explanation (both pictures: www.mytailor.com)

Examples for providing orientation to the user on the state of configuration

The following picture provides two examples of different vendors, how the user in guided through the configuration process, and how the configuration systems indicates the sate of the configuration, i.e. how the user gets an overview where he is, what he has already configured, and which steps are in front and still missing.

Both screen shots show more ore less at the same point in the configuration process. The configuration system on the left side (Dolzer) shows in the right column the single configuration steps and the previous selection. In the middle part of the page, large buttons help to navigate step by step. The one on top guides back, the one at the bottom shows the next layer (step) comes next.

The metaphor of the spindle that is used in the Dolzer web site is a perfect example for an intuitive web site design providing orientation for the user.

Consider in contrast the system on the right picture (Bivolino). While the overall offering of this vendor and the quality of the shirts (i.e. span of fabrics and choices) is much better than in the Dolzer case, the site is rather difficult to navigate and provides less orientation for the user. It is impossible to figure out what has already been chosen, and which steps are still missing. The three drop-down menus are hierarchical organized and change permanently. The only hint on how far the configuration has already become, is the drawing of the shirt. But this drawing doesn't the show the facts very clear, the illustration of the material is more confusing than helpful. In the Dolzer case, the rather small picture in the lower right corner of the web site, shows exactly the state of configuration, including the material in the correct measure.
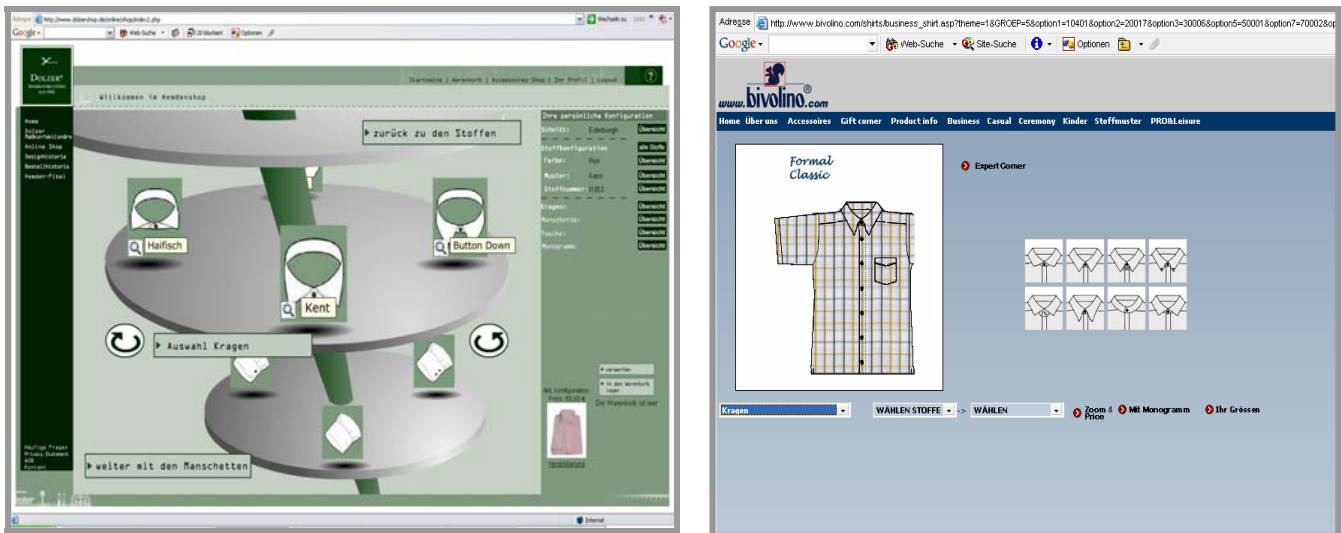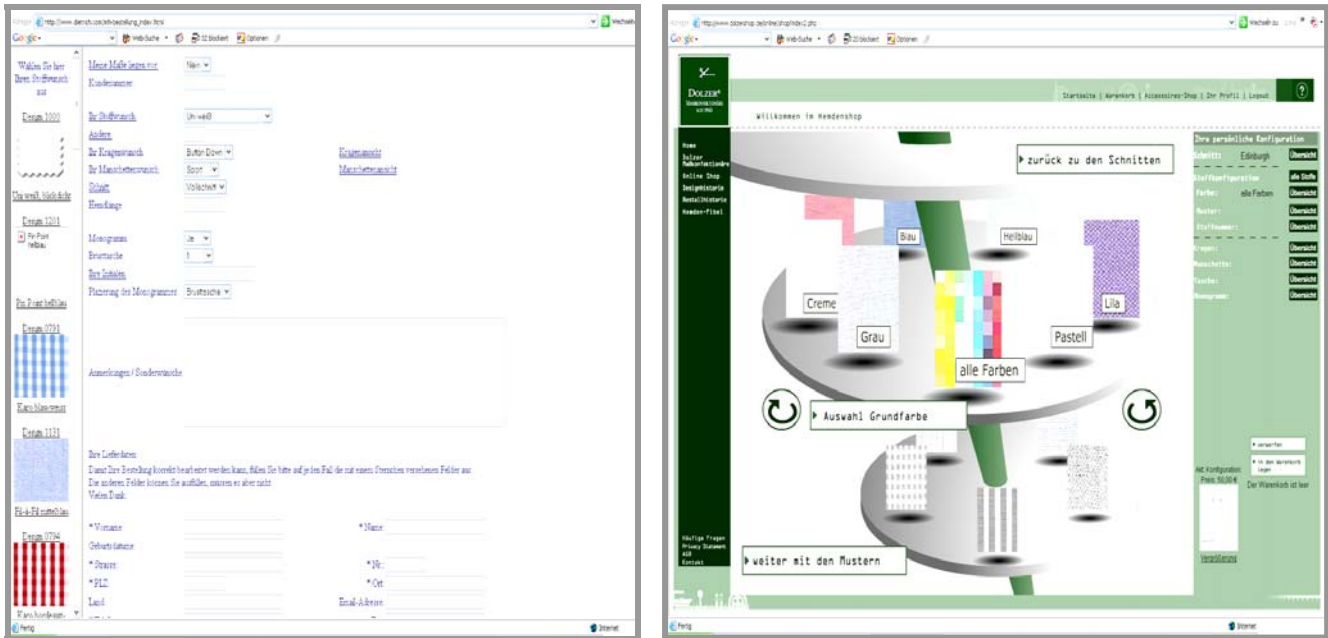

Picture 2: Examples for Orientation (left: www.dolzer.de; right: www.bivolino.com)

Examples for individual access on information

To enable users to access information in their personal way, a configuration system must be able to deal with the different kinds of how information can be presented. The access on information by textual or visual elements is self-evident and well known. The point is to combine these two elements and link them among each other.

The left picture (Dietrich) shows a very textual orientated configuration system. All functional elements are arranged in pull-down menus or are written down by simple links. Even the pictures of the materials are in no connection to the functionality, the user has to figure out the name of the material and than chooses this name out of the menu. Visual oriented people will have difficulties in finding their way though the configuration process.

On the right side (Dolzer), the connection between textual and visual elements is better implemented. Even if a lot of the functionalities are double coated, it gives the user the possibility to choose his way. In the middle of the site are the visual elements. The user can click on these elements to select an option. On the right side is the textual conversion of the functionalities, here the user can see how the elements are described and is also able to choose out of a list of names.

Picture 3: Examples for individual access on information (left: www.dietrich.com, right: www.dolzer.de)

Examples for support

This two examples show how detailed support and additional information can be given. The left screen (MyTailor) shows a very detailed support concept. The systems does not only provide a closer view and description to the single variants, but provides also rather detailed descriptions and a style guide supplementing the advise of a sales person in a offline shop. This degree of information supports also the provision of a feeling of competence and security from the vendor towards the customer.

The configuration system on the right side is clearly arranged, but it doesn't give any support or additional information at all. Such a page is often more increasing the state of confusion than offering support.
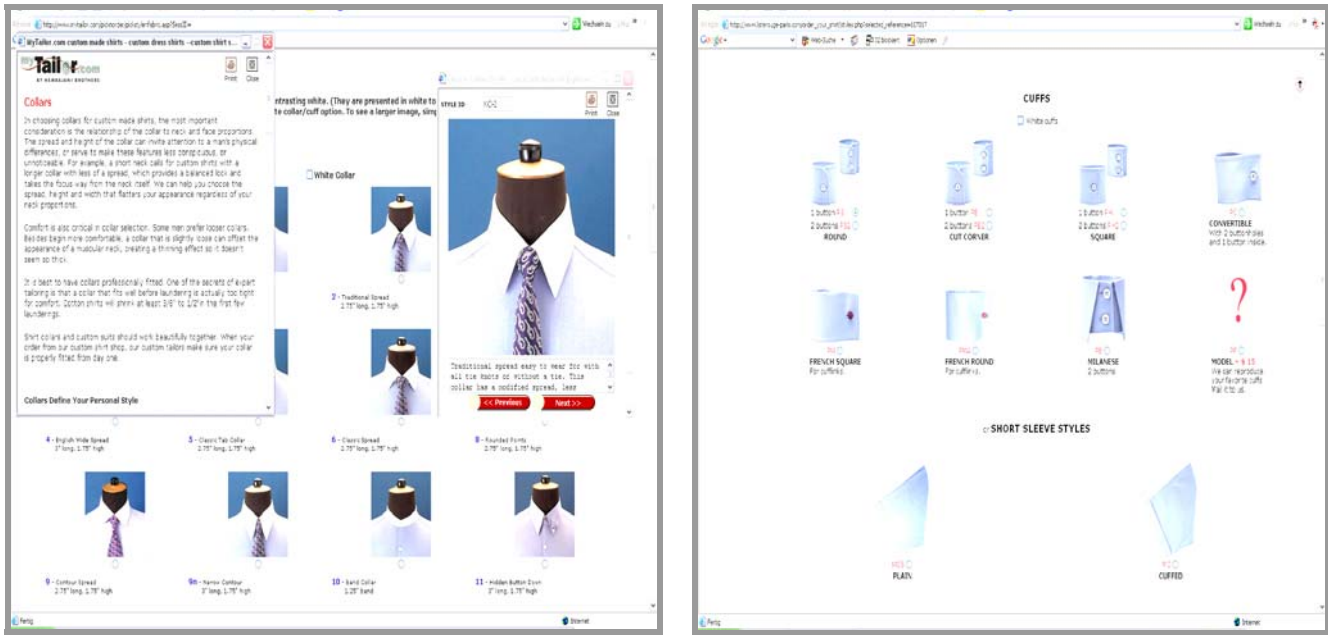
Picture 4: Examples for providing support during the configuration step (left: MyTailor, right: ListRourge Paris)
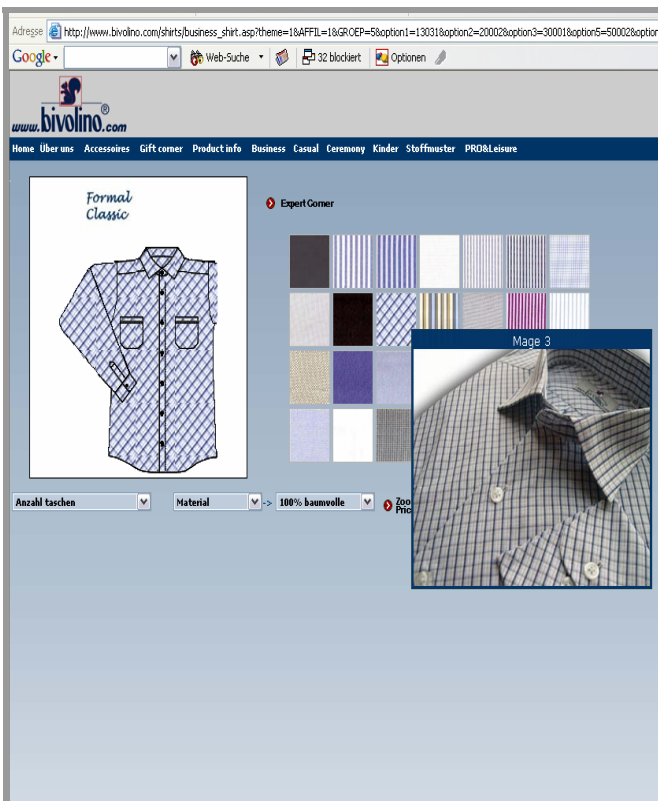
Examples for visualization

There are already a lot of firms offering custom made shirts in the Internet. Every one has its own configuration process although the single elements are more or less the same. The biggest varieties between all site can be found in regard to the presentation of the customized product.

Only very few vendors present the custom made shirt in its overall, customer specific configuration. Most of the vendors just show simple drafts or single components. The left picture (Picture 5, Gert Keller) provides an example which only offers visualization by drafts. To get an impression of the real look and feel, the customer needs a lot of imagination. This visualization can at best help during the configuration process, but will never be in the position to assist the selling process and reduce risk.

The picture at the right (ListRouge) shows photos of the different variants of the collars. This already gives a much better impression on the real look and feel of the product. But even this is a very easy way of visualization only, because the different collars are not even shown in the material which has been selected. A complete picture of the shirt is not available at all.
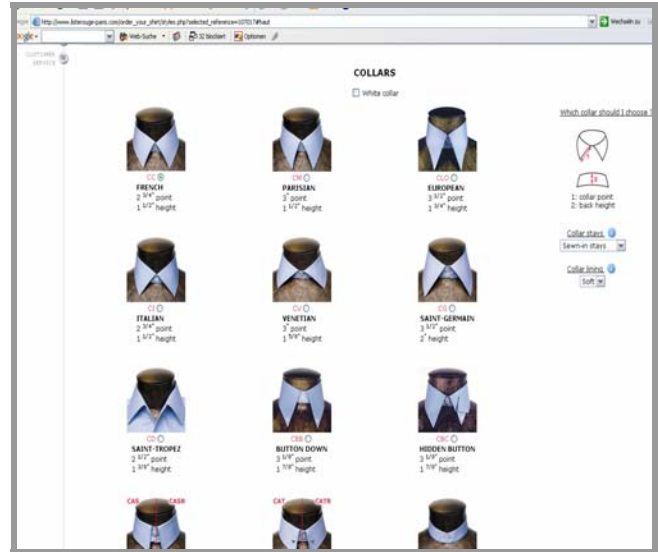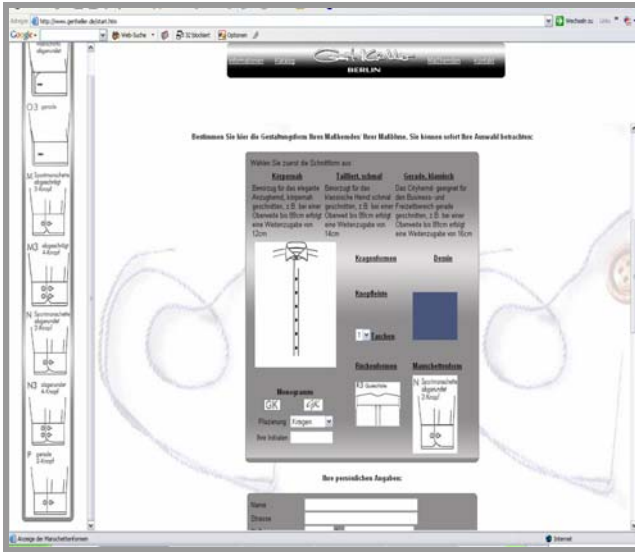
Picture 5: Examples of visualization I (Left: Gert Keller, right: ListeRouge Paris)

Picture 6: Examples of visualization II (Left: Bivolino, right: Dolzer)

A more advanced kind of visualization is shown in Picture 6 on the left (Bivolino). This configuration system combines colored drafts among with standard photographs. This can give a good impression of the selected shirt, but there is also the risk of confusing the customer, seeing a real photo of a shirt, which doesn't have the same characteristics that have been selected in the configuration process. An

other problem with this solution is that the visualization during the configuration process takes only place by adding or removing single elements of the draft.

The picture on the right (Dolzer) provides an example of visualization which is a combination of compound pictures and rendered pictures. Single components like collars, cuffs or pockets are put together by single vector graphics. This vector graphic is afterwards coded by the texture of the material. This is a very good solution to give the customer a real impression of the product. This is even a solution which guides through the configuration, showing each configuration step and its modifications (see again Picture 2, left).

Conclusion and final comparison

It is indeed not an easy task to compare configuration systems on the internet and their functionalities in an objective and generic way. Everybody has, like already mentioned before, an own sense and background knowledge, which shapes the individual perception and handling. The following comparison of all systems in Table 4 has to be interpreted in this way. It shows a comparison of all twelve configuration systems, based on the criteria identified in Chapter 3 in regard to the reduction of risks of the buyer, the usability of the system, and the visualization solution. We have used a rating scale from 0 to 10 to evaluate our impression of the fulfillment of the requirements we have disposed before (Table 3):

Points	Explanation
0	no fulfillment of this requirement at all, feature not available
1-3	feature available, but rather weak solution, sometimes even increasing the complexity of the process from the customer's perspective
4-6	feature available, but only average solution
7-9	feature available, rather good solution, good implementation of this feature creates differentiation of configuration system from its competitors and contributes to positive process experience (flow experience) of the site
10	Excellent solution, best practice

Table 3: Scale and points of evaluation

The final comparison on Table 4 can provide not more than a glimpse, but shows already which differences exist between the systems and where potential for improvement exists. Concluding, we can say that even in an industry that is rather advanced in the implementation of mass customization on the internet, the state of the design of the front ends of configuration systems is rather weak. Many systems are just at the beginning of the stage of development. Compared to other online industries like online banking or general shops for standard goods, configuration systems on the internet are still in a mode 1: rather technical, orientated on getting the job done, but not on creating an experience for the customer and increasing overall satisfaction of the customer by contributing to the process satisfaction.

Configuration System / Degree of Performance	Ravis International	AmericanFit	MyTailor	Bivolino	Liste Rouge Paris	Dolzer	Dietrich	Perleth	Gert Keller	Holl	Stefanelli	The Shirt Tailor
Risk Reduction and Trust Building	4	4	5	6	6	8	4	6	4	7	4	8
Usability/Operability and Self Explanation	6	7	4-8	7	6-8	8	5	7	5	6	4	7
Usability/ Orientation	7	7	5	3	5	9	5	8	4	7	5	9
Usability/ Individual Access on Information	2	3	6	6	5	8	2	6	3	6	3	8
Usability/ Loading Time and Plug-in	10	8	6	7	7	6	6	6	9	6	6	2
Usability/Support	4	2	7	5	4-6	6	3	4	0	5	3	5
Visualization (any kind)	no	Picture	Draft/ Picture	Draft/ Picture	Draft/ Picture	Picture	no	Draft	Draft	Draft/ Picture	Draft/ Picture	Picture
Visualization/ Pre Produced Pictures and Drafts	0	3	5	7	7	8	0	5	4	6	4	8
Visualization/ Compound Pictures of the Configuration	0	0	0	1-3	0	8	0	0	0	0	0	8
Visualization/ Rendered Pictures of the Configuration	0	0	0	0	0	8	0	0	0	0	0	8
Visualization/3-Dimentional Models and Augmented Reality of the Configuration	0	0	0	0	0	0	0	0	0	0	0	0

Table 4: Comparison of the fulfillment of the requirements form the customer's perspective

REFERENCES

1.   Pine, J.B.: *Mass customization*, Boston 1993.

2.   Duray, R., et al.: *Approaches to mass customization: configurations and empirical validation,* in: Journal of Operations Managements, 2000, pp. 605-625.

3.   Tseng, M., Jiao, J.: *Mass customization*, in: *Handbook of Industrial Engineering*, edited by G. Salvendy, New York 2001, pp. 684-709.

4.   Piller, F.T.: *Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter*, Wiesbaden 2003.

5.   Tseng, M., Piller, F.T.: *The customer centric enterprise. An integrative overview on this book*, in: *The customer centric enterprise: advances in mass customization and personalization*, edited by M. Tseng, and F.T. Piller, Berlin et al. 2003, pp. 3-16.

6.   Kotha, S.: *Mass customization: implementing the emerging paradigm for competitive advantage,* in: Strategic Management Journal, 1995, pp. 21-42.

7.   Anderson, D.M.: *Agile product development for mass customization*, Chicago 1997.

8.   Victor, B., Boynton, A.C.: *Invented here*, Boston 1998.

9.   Ahlström, P., Westbrook, R.: *Implications of mass customization for operations management,* in: International Journal of Operations & Production Management, 1999, pp. 262-274.

10.   Huffman, C., Kahn, B.E.: *Variety for sale: mass customization or mass confusion,* in: Journal of Retailing, 1998, pp. 491-513.

11.   Lee, S.C.-H., Barua, A., Whinston, A.B.: *The complementarity of mass customization and electronic commerce,* in: Economics of Innovation & New Technology, 2000, pp. 81-110.

12.   Reichwald, R., Piller, F., Moeslein, K.: *Information as a critical success factor for mass customization*, in: *Proceedings of the ASAC-IFSAM 2000 Conference*, Montreal, Quebec 2000.

13.   Davis, S.: *Future perfect, Reading: Addison-Wesley*, 1987.

14.   Kotler, P.: *From mass marketing to mass customization,* in: Planning Review, 1989, pp. 10-13.

15.   Toffler, A.: *Future shock*, New York 1970.

16.   Picot, A., Reichwald, R., Ralf, W.: *Die grenzenlose Unternehmung*, Wiesbaden 2003.

17.   Helander, M.G., Khalid, H.M.: *Customer needs in web-based do-it-yourself product design*, in: *Proceedings of the 10th Anniversary of M.Sc. Ergonomics International Conference*, Lulea, Sweden 1999.

18.   Tseng, M., Du, X.: *Design by customers of mass customization products,* in:CIRP Annals, 1998, pp. 103-106.

19.   von Hippel, E.: *Economics of product development by users: the impact of "sticky" local information,* in: Management Science, 1998, pp. 629-644.

20.   Hibbard, J.: *Assembly online: the web is changing mass production into mass customization,* in: Information Week, 1999, pp. 85f.

21.   Zipkin, P.: *The limits of mass customization,* in: Sloan Management Review, 2001, pp. 81-87.

22.   Franke, N., Piller, F.T.: *Key research issues in user interaction with configuration toolkits,* in: International Journal of Technology Management (IJTM), 2003, pp. 578-599.

23. Gilmore, J.H., Pine, J.B.: *Customization that counts*, in: *Markets of one*, edited by J.H. Gilmore, and J.B. Pine, Boston 2000, pp. vii-xxv.

24. Pine, J.B., Peppers, D., Rogers, M.: *Do you want to keep your customers forever?,* in: Harvard Business Review, 1995, pp. 103-114.

25. Vandermerve, S.: *Personalization creates experiences, Keynote "First European Personalization Summit"*, London 2000.

26. Riemer, K., Totz, C.: *The many faces of personalization - an integrative overview of mass customization and personalization*, in: *The customer centric enterprise: advances in mass customization and personalization*, edited by M. Tseng, and F.T. Piller, Berlin et al. 2003, pp.

27. Fama, E.F., Jensen, M.C.: *Separation of ownership and control,* in: Journal of Law and Economics, 1983, pp. 301-325.

28. Weston, R.: *Web automation,* in: PC Week, 1997, pp. 76.

29. Bourke, R.: *Product configurators: key enabler for mass customization - an overview,* in: Midrange Enterprise, 2000, pp.

30. Williamson, O.E.: *Markets and hierarchies: a study in the economics of internal organization*, New York 1975.

31. Williamson, O.E.: *Transaction cost economics*, in: *Handbook of industrial organization*, edited by R. Schmalensee, and R.D. Willig, Amsterdam 1989, pp. 135-182.

32. Csikszebtmihalyi, Mihaly: *Das Flow-Erlebnis, Stuttgart, Klett Cotta 1999*

33. Bauer, Hans H., Gertner Mark, Borrmann Ulrike*: die Erklährung des Nutzungsverhaltens in elektronischen Medien mit hilfe der Flow-Theorie, in Marketing ZFP, 23 Jg. (2001), Nr.1, pp. 17-30*

34. Rogoll, T., Piller, F.T. Konfigurationssysteme für Mass Cusomization und Variantenproduktion; think consult, München 2002

35. Piller, F.T.: *Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter*, Wiesbaden 2003.

36. Pixelpark: Usability Findings, in: Newsletter Marketing Research, Berlin: Pixelpark AG 2000.

37. Tanja Diezmann: *Navigation and Usabilty, in Usability Nutzerfreundliches Webdesign*, Springer 2002

38. Teresko, J.: *Mass customization or mass confusion,* in: Industry Week, 1994, pp. 45-48.

REDUCING COMPLEXITY FOR CUSTOMERS
BY MEANS OF A MODEL-BASED CONFIGURATOR AND
PERSONALIZED RECOMMENDATIONS

Thomas Leckner, Rosmary Stegmann, and Johann Schlichter

Technische Universität München
Institut für Informatik
Lehrstuhl Prof. Dr. Johann Schlichter
Boltzmannstr. 3
D-85748 München, Germany
Email: {leckner, stegmann, schlichter}@informatik.tu-muenchen.de

During the last years, the Internet has become a medium for innovative business strategies such as «mass customization», the mass production of individually customized goods and services. In mass customization manufacturers try to offer both variety and customization. To realize this business strategy manufacturers normally provide so-called online configurators which enable customers to interactively specify and adapt a product in accordance to their individual preferences. But the more possibilities such configurators offer, the more complex the whole specification process becomes for the customer. This paper introduces the prototype of a Web-based configurator for products with many degrees of freedom that supports the customer by personalized recommendations. Both the tool from the user's point of view and the underlying recommendation methods are described.

Significance: The individualization of products with many selection possibilities and interdependencies between these depends heavily on suitable tools that help to support the customer during the complex process of specifying preferences and interactively configuring a product. This paper describes a framework for an appropriate approach enhanced by personalized recommendations.

Keywords: Mass Customization, Product Configuration, Customer Support, Product Recommendation.

1. INTRODUCTION AND MOTIVATION

Whenever an enterprise decides to offer individualized products to its consumers, it must cater to its customers' individual needs. Communication and interaction with the customer is critical, because the customer himself defines the product he wants by expressing his individual product wishes. The communication and interaction with the consumer can clearly be facilitated by Internet tools. Initial technical approaches for this purpose are online configurator tools (Sabin and Weigel, 1998; Felfernig et al., 2001), which can be found, for instance, on the homepages of the large automakers, computer vendors, and in the fashion industry.

In the **interdisciplinary research effort SFB 582** "Marktnahe Produktion individualisierter Produkte" («production of individualized products close to the market», http://www.sfb582.de) we focus on the individualization of **complex mechatronic products,** such as cleaning robots, cooling units, household appliances, and personal medical equipment. The SFB582 project is joint research of the Mechanical Engineering, Business Administration, and Computer Science Departments of the Technical University of Munich, Germany, that investigate aspects of specifying and manufacturing personalized products in the framework of a suitable **Mass Customization** (Piller, 2001) business strategy. The Computer Science subproject investigates the issue of **customer support** through the use of adequate tools and methods during the configuration of individualized products (Leckner et al., 2003).

One of the major problems with the configuration of complex products is that the process of configuration can become extremely tedious to the customers if they are forced to deal with the entire complexity of product variation. This may simply be because the product has so many components and degrees of freedom (see Section 2) that it takes a very long time to work through all of them manually. On the other hand, customers are often aware of only the specific functionality which they require, but are not aware of which attribute values to set and which components to choose in order to achieve this functionality. They are often also unaware of the interdependencies between components and functionalities. In the worst case customers do not even know their own preferences (Wind et al., 2002; Blecker et al., 2004). Thus, recommendations generated either automatically or by other consumers via a customer community (Leckner, 2003) might be helpful. Any customer who is not assisted in solving problems during manual configuration may stop and buy the product elsewhere.

The solution we have developed and which is described in this paper is a **Web-based platform** that integrates a comfortable configuration tool and automatically generates product recommendations on the basis of user profiles. Generating recommendations for complex products creates a particularly difficult problem. To «build» the recommendation from a set of product components and attribute values for each customer individually requires not only interaction with the user profile but also with a fine-grained product model that allows for constraint checking in order to recommend only valid configurations. It should be obvious that this dynamic building of a product recommendation is much more difficult than remaining with a recommendation of a complete and static product, which is currently done in some online shops.

The focus of this paper is to present our approach of a configurator tool for complex products, which is enriched through different customer support mechanisms. We focus on requirements, functionalities, and the user interface of our configurator tool and discuss two techniques for automatically generating recommendations we have implemented. In Section 2 we explain which requirements we have defined before starting to implement our configurator tool. In Section 3 we present our configurator tool, describe its functionality and its «look and feel» for the customer. Section 4 is about the generation of product recommendations, where two specific recommendation techniques are introduced in more detail. Section 5 concludes the paper and gives some prospects for future work.

2. REQUIREMENTS AND PROBLEMS TO SOLVE

One significant difference between our configurator tool and existing systems on E-Commerce homepages can be seen in the complexity of the products. Our idea in the SFB582 project is to give the customer complete product design freedom. This idea leads to specific requirements for our configurator tool, which are described in this section. Section 3 shows how these requirements are met by the design of our tool.

2.1 Products with many degrees of freedom for the customer

In usual configurators the customer is somehow guided through a rather small set of product variables and can select from a set of predefined variants for each of these variables. In contrast to this, our aim is to realize a system where the customer has enhanced and even more complex possibilities to adapt the product in accordance to his individual preferences and needs. In consequence the configurator tool rather aims at presenting to the customer the whole physical structure of a complex product with all its components, sub-components, and all properties of those components.

Therefore, the first fundamental requirement for our Web-based configurator tool was the capability to handle products with many **degrees of freedom**. With the term «degree of freedom» we describe those parts of the product, which can be modified by the customer (see Section 4.1).

Additionally we also had to determine methods of **presenting such complex products to the customer** in an appropriate manner. On the one hand, the customer needed an overview of the product and its structure. This necessitates a customer's ability to navigate through the multitude of product possibilities. On the other hand, we had to enable the customer to influence the various degrees of freedom, ranging from a change in the product's structure to the assignment of values for different types of component attributes. Also, some sort of configuration checking was necessary to evaluate the viability of the customer's choices.

2.2 Additional support for the customer

The configuration task for complex products with lots of degrees of freedom can, of course, become quite tedious and time-consuming for the customer. Moreover, the effect of many degrees of freedom can discourage the customer due to his limited knowledge and experience in developing and customizing products. In fact, too many choices may confuse the customer (Helander and Jiao, 2002).

In order to overcome the complexity during the configuration task, we defined additional support mechanisms as another important requirement for our configurator. But in contrast to the approach of an *advisory system*, introduced by Blecker et al. (2004), we did not want to hide the whole complexity of configuration from the customer. Instead, the customer must manually configure the complex product, but is extensively supported by additional functionality.

For this purpose, on the one hand we embedded **statistical information** into the configurator tool, which makes it necessary to present, for example, how often a certain value for a degree of freedom has been chosen in the past. On the other hand we enhanced the configurator tool with **personalization techniques** in order to give recommendations to the customer in an automatic way. Recommendations are given in various granularities (Stegmann et al., 2003) and are generated by different methods. Here we especially focused on user profile-based filtering. Thus, the capability to acquire, store, and analyze user profiles had to be incorporated in our configurator tool. Our work on profile acquisition and management will not be described in this paper, however.

Additionally we wanted to overcome shortcomings of single filtering approaches (Melville et al., 2002) through the dynamic combination of different filtering methods. We therefore included in our configuration tool the Pipelined Filter Combination approach that is described in more detail in Renneberg and Borghoff (2003).

2.3 Adaptability to different product models

A final important requirement for our Web-based configurator tool was openness towards product models. We did not want to realize a tool which was restricted to a specific kind of product. Instead our aim was to build a model-based tool which could **dynamically adapt to an underlying product model**. In our approach the user interface of the configurator was supposed to be generated «on the fly», while loading the respective product model. The **product model** itself describes the product's structure and all the degrees of freedom through the use of conceptual building blocks, which are defined in a general product meta model (Janitza et al., 2003; Leckner and Lacher, 2003).

To provide the necessary adaptability we decided to realize the Web-based configurator tool by using the Java Servlet Technology, which allows us to generate web pages dynamically in accordance to the product model and the customer's interactive configuration session.

3. THE MODEL-BASED CONFIGURATOR TOOL

In this section we present our configurator tool and describe how it meets the requirements listed above. A potential customer may use the configurator to specify his desired product manually. It is based on a detailed model of the product, its components, and degrees of freedom. The tool integrates a product and component catalog for easy browsing and visualizes personalized product and component recommendations. Additionally, it provides various customer support methods. In this section we explain these aspects in more detail, mainly from a user's point of view.

3.1 Visualizing the product structure and degrees of freedom

The underlying **product model** of our configurator tool is designed from a customer's point of view rather than from a manufacturing point of view. This means we mainly model those components, features, and functionalities of the product that can be manipulated by and are of particular interest to the customer. Nevertheless we want to provide a complex product model to enable customers to individualize the product (see Section 4.1). Therefore it is necessary to choose a visualization method that facilitates customer usage, but still allows for a complete overview of the product structure and product details.

Navigation through the product

The product model, representing a complete product, is structured as a tree of product components. In our tool the customer can navigate through this component tree and view and manipulate details of each component in separate parts of the browser. The navigation is supported by the **navigation tree** and the **navigation bar** (see Figure 1).
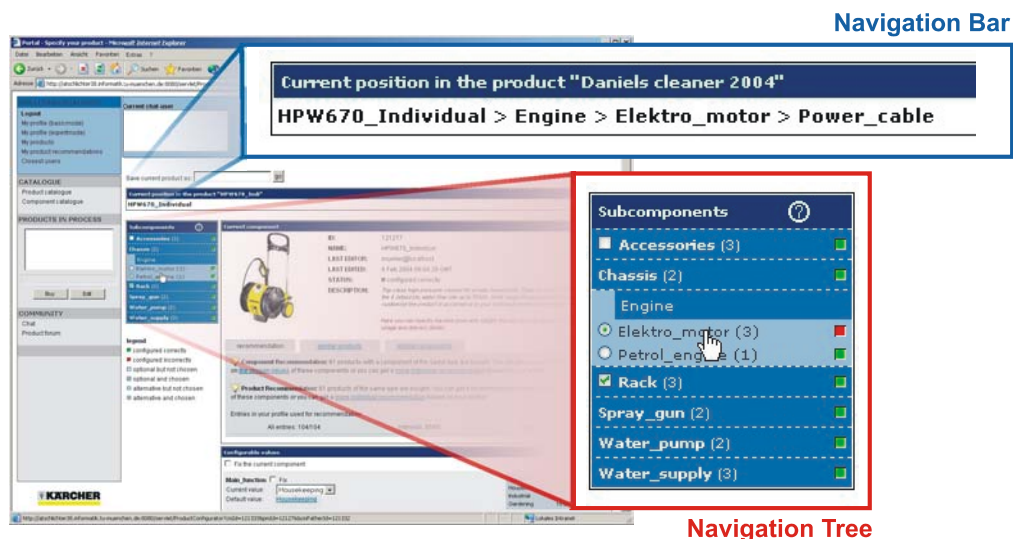

Figure 1. Navigation tree and navigation bar

With the **navigation tree** the customer can browse through the subcomponents of the current component, while the **navigation bar** on top indicates the actual position in the product as a component path. It allows for easy navigation in the component hierarchy and is helpful for navigating back to parent components or to the root component. The customer's navigation through the product can be performed quite easily by clicking on a component name either in the navigation tree or in the navigation bar. All component names within the navigation tools are labeled with hyperlinks, which are created automatically while loading the respective Web-page. Even the Web-page itself is created dynamically in accordance to the actual *product model ID*, *component model ID* and the customer's *user ID*.

Furthermore, components may be optional or alternative in our product model, which is considered as degree of freedom and is also visualized within the navigation tree. Alternative components are represented by radio buttons; optional components by check boxes, which the customer can use to choose components (cf. Figure 1). After the name of each component the number of subcomponents is displayed. A little box behind each component indicates whether this component is configured correctly (green color) or incorrectly (red). This is necessary to visualize violated constraints between degrees of freedom. Only if all values of a component are legal is the component marked green, and only if all components are configured correctly will the entire product have a green label «configured correctly».

We have mentioned before that our tool is not bound to one product model type only. The product and component catalog helps the user to browse different product types and select a specific type and model as the basis for his individual adaptations. This also reduces complexity for the customer, since it does not require the customer to specify a complete product from scratch.

Details of a component model

Let us now concentrate on the **detail view of a product component** which a customer may choose in the navigation tree. The detail view is subdivided into three parts (cf. Figure 2): (1) The top part shows the name and the component or product ID, a picture and a short description of the component or product as well as the last date of configuration, the person who has configured it, and the status of the correctness of the configuration. (2) The middle part describes and provides link tabs for different kinds of support methods which can be obtained. (3) The bottom part shows the degrees of freedom of this component, e.g. all attributes and types of possible values such as selection lists or selection ranges.
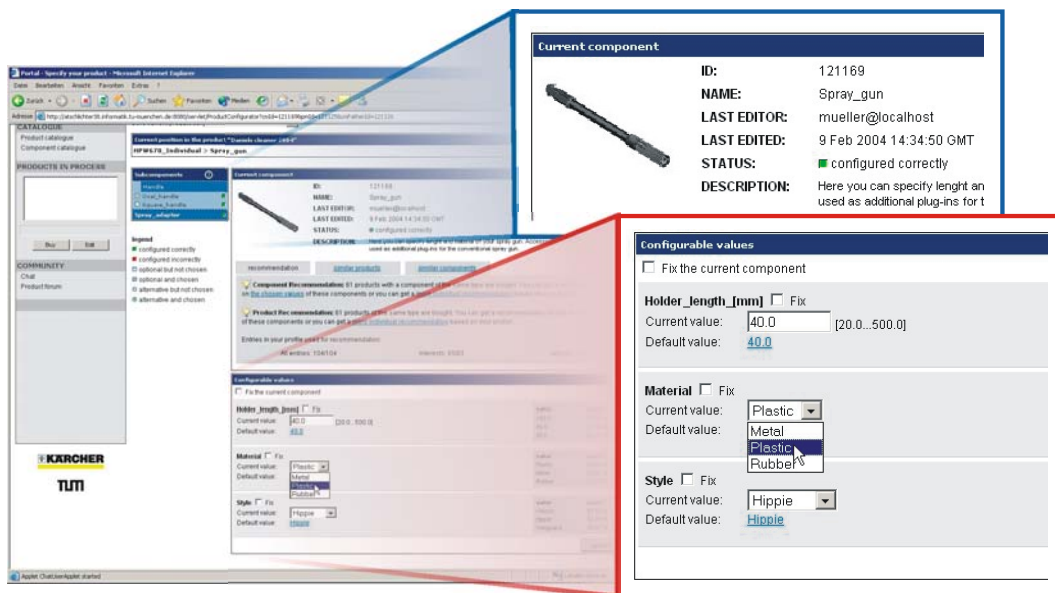

Figure 2. Detail view of a component

The customer can change each attribute value manually, while maintaining an overview of possible value ranges. To send the changes to the server the customer must click the button «update values» or press the return key.

3.2 Additional support for the customer

In order to support the customer's decisions during the configuration process we have integrated not only a recommender system (Section 3.3) but also additional functionalities into our configurator tool:

Browsing the product and component catalog manually can be quite tedious, especially if the customer has already set certain values of a component and wishes only to quickly review the configuration used by other customers for the same

component. For this case we integrated the functionality «**similar components**» and «**similar products**». This functionality can be seen as a kind of «smart browsing» of the catalog in order to avoid the need for the customer to search the entire catalog manually. If the customer has set some values for i.e. the component *spray gun* and clicks on «similar components» he obtains a list of *spray guns* which have been configured by other customers in the past and which are similar to the attribute values chosen for his *spray gun* so far (see Figure 3). For each similar component that has been found the degree of similarity is indicated by percentages. The customer can click one of these components and a new window opens. In this window (bottom of the right side in Figure 3) the customer can browse the component and accept it for his own product, if he likes, by pressing the button «apply».
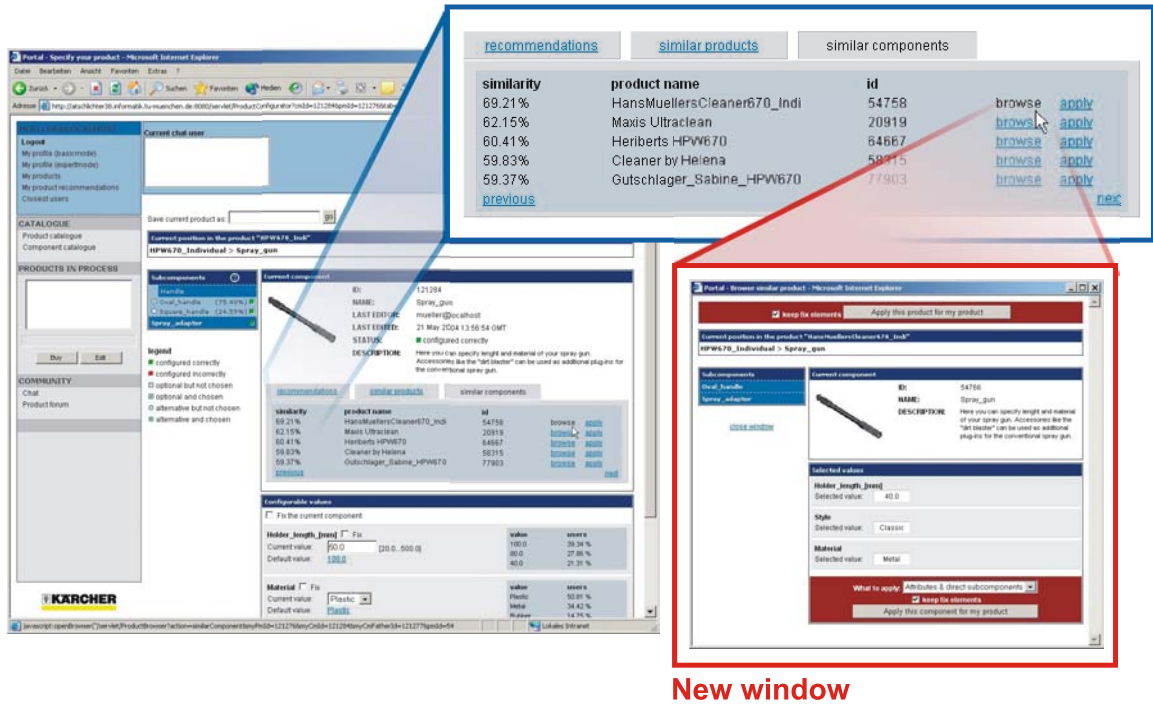

Figure 3. Similar components

For each degree of freedom (i.e. either components or attributes) the user can get an **overview of other customer preferences**. Figure 4 shows two examples: (1) Percentage values illustrate alternative components previously selected by other customers in the navigation tree. (2) The top three of all formerly configured values are shown for an attribute model in the detail view of a component. This functionality helps solve uncertainties with respect to concrete value and component choices.
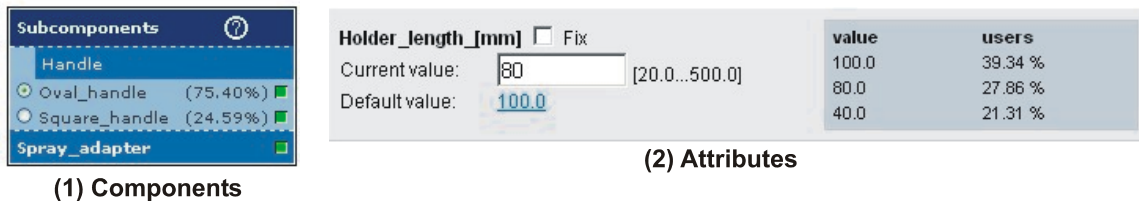

Figure 4. Indication of customer selection preferences

3.3 Visualization of recommendations

As we have mentioned above we also support the customer by providing recommendations. Recommendations can be given both for a complete product and for specific degrees of freedom alone. Whenever a recommendation is provided, the customer can decide whether or not to apply it, by clicking on the respective links of the configurator tool.

Recommendations may have different effects in our system and therefore we also need different ways of visualizing them, which are illustrated in Figure 5: If the effect of a recommendation is a **concrete value setting for a degree of freedom**, the color of that attribute's or component's name will be changed and the former value of the degree of freedom will be indicated below the newly set value. If the effect is the **removal of values for the degrees of freedom** (e.g. hiding

alternative components from the navigation tree, because they do not seem to fit the customer's preferences) the number of visible degrees of freedom will be indicated enabling the customer to return to the full model when wanted. The same is true, if the recommendation causes the restriction of possible attribute values (see Figure 5).

Figure 5: Visualization of recommendations

Note that each way of visualizing recommendations allows the customer to determine whether to continue or return to the former values, larger visible value ranges, or the full range of components, etc.

4. PRODUCT MODEL AND RECOMMENDATIONS

In Section 3 we have introduced some aspects of the user interface and the functionality of our configurator tool. Important characteristics of the tool are the possibility for creating and visualizing recommendations, and the fact that the whole tool can dynamically adapt to different product types. In the following we will explain the data structures for realizing our product model, especially with regard to the generation of recommendations. Also we will discuss the two filtering mechanisms we have implemented so far and briefly sketch out important challenges during the generation of recommendations for a configurable product.

4.1 Modeling of products, components, and attributes

As we have seen in Section 3.1 the product model is structured as a **tree of product components** each of which can in turn consist of a set of components, some of which may be optional or alternative. The conceptual building blocks, used to model the whole product spectrum (and describing, for example, how components can be connected in the component tree), are defined by a product meta model. More details about our meta model can be found in Leckner and Lacher (2003).

When examining the data structure of the component tree, the existence of an especially marked root component model can be seen. In addition, each component model contains a list of pointers to its subcomponent models, also stating if a subcomponent model is optional or not. The data structure of an alternative component model is implemented as a special extension of a normal component model object. On the database layer, every component model has a unique model ID, and there is a database table, where the relationships between components and subcomponents are stored. This table also manages the optionality of subcomponents. For alternative component models a special flag is set in the database table, where details about component models (like name, description, editor, etc.) are stored.

Furthermore, each component model is characterized by a set of **attribute models**, which represent the properties of the concerned component. The relationships between component models and attribute models are realized in a manner similar to that of the component-subcomponent relationships. The only difference is that attributes can never be optional. Besides

we have different types of attribute models, namely we have implemented *numeric interval* attributes and *enumerated set* attribute models so far. For future work we intend to implement additional attributes of varying types.

The values of the attribute models and the selections for optional/alternative component models are the **degrees of freedom** in our product model. These must be assigned by the customer (see Section 3.1) and/or by the recommender system (see Section 3.3). As seen in the configuration modeling concepts described by Männistö et al. (2001), our approach allows every degree of freedom to have a *range of valid values* (e.g. the upper and lower boundaries of a numeric interval attribute), a *selected value*, a *default setting* and also *constraints* between different degrees of freedom. These **constraints** are modeled and evaluated by using the Protégé Axiom Language (PAL), a superset of first-order logic which can be used to express constraints (see: http://protege.stanford.edu/plugins/paltabs/PAL_tabs.html) and to make logical queries about a knowledge base. We decided to model constraints in PAL because we used the open source platform Protégé (see: http://protege.stanford.edu/) to build our product meta model and to design initial product models in a graphical and user-friendly manner.

The initial product, component and attribute models, designed within the Protégé tool are called **creatorModels**. All models, made within the configurator tool belong to their respective creatorModel and therefore have a reference stored to it. This is necessary in order to distinguish between initial models and customer configuration models.

Each model has a reference to a **baseModel**, from which it was derived. The idea of this approach is illustrated for an attribute model of type *numeric interval* in Figure 6: A_0 is the initial creatorModel, while A_X and A_Y are models, which have been adapted by customers X and Y. First, customer X used the initial creatorModel A_0 to specify his attribute model, while customer Y started with the configuration settings of customer X.
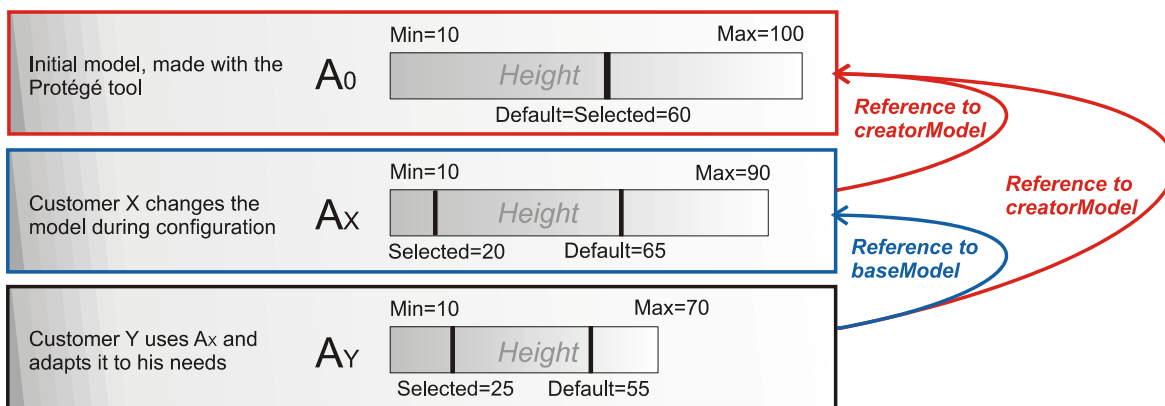

Figure 6. CreatorModel and baseModel

If customer Y had not changed anything, no new attribute model would have been created. In fact there would be just the attribute A_X for both customers. But customer Y changed the value and therefore the new attribute model A_Y was created, which has a reference to A_X. To clearly identify the initial source model more quickly, a direct reference from A_Y to the creatorModel A_0 is stored.

4.2 Generation of recommendations

Based on the product modeling described above, our configurator tool can create recommendations for the customer. Whenever giving a recommendation, a new **restricted product model** is created that refers to the baseModel, from which the recommendation was created and to the initial creatorModel. The recommended model always is a subset of the original baseModel (as seen in Figure 6). Imagine that A_Y is a restricted model for A_X which was created by a filtering method. The possible value ranges for the numeric interval attribute are reduced in A_Y compared to A_X. Also the default settings and the selected values are changed in accordance to the results of the recommender system.

We have so far implemented history-based filtering and collaborative filtering to create recommendations, which are currently part of our configurator tool prototype and which are described in the following.

History-Based Filtering

To achieve history-based filtering, the customers' buying histories are analyzed. This approach assumes that customers are likely to select those values for the degrees of freedom, which the majority of other customers have previously selected. For the **analysis of the buying history** of a specific model M, the reference to its creatorModel M_0 (see Section 4.1) plays a crucial role. It allows to query the database for those models $\{M_i\}$ ($0 \leq i \leq n$ and n is the total number of models stored in

the database) which have a reference to the same creatorModel M_0 like M. The degree of freedom values which have been overwhelmingly selected thus far by others are then recommended to the customer.

Collaborative filtering

The collaborative filter is similar to the history-based filter in that previously purchased products are again analyzed. But as opposed to history-based filtering not all products are weighted equally. A special value for a degree of freedom will be weighted according to the similarity of the previous buyer to the current customer. This means that in addition to the analysis of the previously purchased products there is also an analysis of customer profiles. The higher the customer similarity, the greater the importance of the products purchased. The main idea of this filter is that similar minded customers will tend to buy similar products (cf. Breese et al., 1998).

The **analysis of the similarity of customer profiles** is performed by different database queries on the user profiles. In addition to demographic data and information about the customer's name and address, information about the customer's interests and ratings for different products are stored in the profile. Both interests and ratings are stored as normalized float values of the interval $[0\ldots1]$. We could also say, that such a normalized value expresses the probability of the customer appeal or interest in the rated object.

On the basis of such normalized values the similarity between two profiles A and B can be calculated as follows:

$$\text{Formula 1.} \qquad similarity(A,B) = 1 - \frac{\sum_{i=1}^{n} abs(A(i)-B(i)) \cdot weight(i)}{\sum_{i=1}^{n} weight(i)}$$

where *A(i)* are rated interests in the user profile of customer A and *B(i)* are rated interests in the profile of user B. Whenever a customer does not explicitly give a rating, the standard value chosen here is *0.5*. Also the different elements *A(i)* and *B(i)* can be weighted by *weight(i)* in accordance to their importance and relevance for the collaborative filter. Additionally the *weight(i)* is set lower when less customers have made an explicit selection for that specific element *i*. In the end we achieve a similarity measure $similarity(A,B) \in [0\ldots1]$ of two customer profiles *A(i)* and *B(i)*, which can be used to weight the relevance of a previously bought product with respect to the referred customer (cf. Karacapilidis and Leckner, 2004).

4.3 Challenges when recommending individualized products

In Section 4.2 we have introduced methods for generating a recommendation for a single degree of freedom. These methods are also used to generate recommendations for products of the shelf, like for instance in online book stores. In the scenario of configurable products, however, we cannot simply recommend a value for one degree of freedom, without taking into consideration the other degrees of freedom. **Interdependencies between degrees of freedom** might be possible, meaning constraints might be violated whenever changing the value of a single degree of freedom. Therefore, when generating recommendations the whole product model must always be considered. In consequence, the input and output of every recommender is a complete product model, even if only the value of one specific degree of freedom has to be changed.

Obviously, when the whole product model is an input for the recommender, a possibility to **mark those parts of the product model the recommender shall change** is necessary. In our approach we use a specific path-description to exactly point to the element in the hierarchical product model the recommender shall change. Additionally, we have **fix-markers** in the product model to allow the customer to explicitly mark degrees of freedom which must not be changed by the recommender system. In Figure 2, for instance, fix-markers can be seen right behind the attribute names.

In a nutshell, there exist different external conditions, which have to be taken into account whenever generating a personalized product recommendation: (1) Constraints between degrees of freedom, (2) markers for those degrees of freedom which may be changed, and (3) markers for those degrees of freedom that are fixed. There are also differences in the quality of a recommendation: In some cases the system may decide very clearly what to recommend. But in other cases it might be difficult for the recommender to decide, because all possible values are about equally good or bad for the customer. In order to take care of such external conditions we have developed the **concept of scoring functions** for degrees of freedom. This approach is described in detail in Renneberg et al. (2004) and allows us, to score the possible values for all degrees of freedom. On the basis of these scoring functions a maximum score for the whole product can be calculated, leading to a valid recommendation. Additionally the scoring functions constitute the combination of different filtering methods, which can be specialized for specific parts of the product model (Renneberg and Borghoff, 2003).

5. SUMMARY AND PROSPECTS

In the previous sections we have given an introduction to the prototype of our Web-based configurator tool. The tool is completely model-based and is appropriate for products with many degrees of freedom and a high complexity for the customer. In addition, the system is enhanced by different support functionalities, especially by methods for **generating personalized recommendations**. In this paper we have presented two filtering methods that have already been integrated into our configurator. These methods already help reduce configuration complexity for the customer to a great deal. However, it would be desirable to realize additional collaborative, as well as content-based and rule-based filtering methods. These should be more specialized on certain parts of the product model and/or on specific elements of the user profile. The combination of different filtering results can be facilitated by the concept of product scorings, which also should be further investigated.

An important condition for the generation of all kinds of personalized recommendations is the existence of **comprehensive customer profiles**. Only with explicit information about the customer we can generate proper recommendations. Therefore, a main focus of our future work is the modeling and analysis of complex user profiles and new and innovative methods of customer profile acquisition. We also intend to use methods from natural language processing to improve and simplify the customer profile management.

Additionally, community-based methods for supporting the customer during the configuration task seem promising (cf. Leckner, 2003). In future research we will, therefore, also concentrate on the idea of **collaborative product configuration** where customers can synchronously and asynchronously interact with other customers or even with professional product designers to help them during the design task through recommendations and an exchange of experiences about the product.

Finally, we would like to point out that our configurator tool has initially been designed for integration into a more **general framework of customer interaction**. The characteristics of such a framework would include the support of multi-channel customer interaction and its integration into a larger order management system in such a way as to provide a detailed product description for manufacturing and service processes. Also the long term management of product and configuration knowledge would be an important issue for such a framework (Männistö et al., 2001). Since the focus of our future work is on customer profiles and collaborative product configuration, we will not realize these integration aspects ourselves, although we think it would be an interesting application field for our tool. Another issue related to the more general framework affects the integration of additional product-related information into the configurator tool, like the product's prize and date of delivery. It would be interesting to investigate, whether this information, which has to be provided by other modules than the configurator, together with a further improved user interface for the configurator and a better visualization of the actual product (e.g. by using CAD) might have an impact on the customer's buying decisions.

6. REFERENCES

1. Blecker, T., Abdelkafi, N., Kreuter, G., and Friedrich G. (2004). Product Configuration Systems: State-of-the-Art, Conceptualization and Extensions. In: Hamadou, A.B. / Gargouri, F. / Jmaiel, M. (eds.): Génie logiciel & Intelligence articielle (MCSEAI 2004), Soussee, Tunisia.

2. Breese, J., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In: Proc.of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), Wisconsin, USA.

3. Felfernig, A., Friedrich, G., and Jannach, D. (2001). Conceptual modeling for configuration of mass customizable products, Artificial Intelligence in Engineering, Vol. 15 (2001), 2, pp. 165-176.

4. Helander, M. and Jiao J. (2002). Electronic Product Development (ePD) for Mass Customization, In Proc. of the 6th International Conference WWDU 2002, Berchtesgaden, Germany.

5. Janitza, D., Lacher, M., Maurer, M., Pulm, U., and Rudolf H. (2003). A product model for Mass Customization products. In Proc. of the 7th International Conference on Knowledge Bases Intelligent Information & Engineering Systems (KES2003), Oxford, United Kingdom.

6. Karacapilidis, N. and Leckner T. (2004): A Recommendation Based Framework for Online Product Configuration, In Proc. of the <u>6th International Conference on Enterprise Information Systems</u> (ICEIS2004), Porto, Portugal.

7. Leckner, T. (2003). Support for online configurator tools by customer communities, In <u>Proc. 2nd International World Congress on Mass Customization and Personalization</u> (MCPC'03), Munich, Germany.

8. Leckner, T., Koch, M., Lacher, M., and Stegmann R. (2003). Personalization meets Mass Customization – Support for the Configuration and Design of Individualized Products, In <u>Proc. of the 5th International Conference on Enterprise Information Systems</u> (ICEIS2003), Angers, France.

9. Leckner, T. and Lacher, M. (2003). Simplifying configuration through customer oriented product models, In <u>Proc. of the 14th International Conference on Engineering Design</u> (ICED 2003), Stockholm, Sweden.

10. Männistö, T., Soininen, T., and Sulonen, R. (2001): Modeling Configurable Products and Software Product Families, In <u>Proc. of the Software Configuration Workshop</u> (SCM-10) of ICSE01, Toronto, Canada.

11. Melville, P., Mooney, R., and Nagarajan R. (2002). Content-Boosted Collaborative Filtering for improved recommendations, In <u>Proc. of the 18th National Conference on Artificial Intelligence</u> (AAAI-02), Edmonton, Canada.

12. Piller, F. (2001). <u>Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter</u>, Gabler Deutscher Universitäts-Verlag, Wiesbaden, Germany, ISBN: 3824478749.

13. Renneberg, V. and Borghoff, U.M. (2003). Pipelined Filter Combination in Product Personalization, In <u>Proc. of 10th International Conference On Human-Computer Interaction</u> (HCI International 2003), Crete, Greece.

14. Renneberg, V., Leckner, T., and Stegmann R. (2004): Recommending Personalized Products Based on Product Scorings, To appear in: <u>Proc. of the Workshop on Configuration 2004</u> held in conjunction with 16th European Conference on Artificial Intelligence (ECAI-04), Valencia, Spain.

15. Sabin, D. and Weigel, R. (1998). Product Configuration Frameworks – A Survey, <u>IEEE Intelligent Systems & their applications</u>, July/August 1998 (Vol. 13, No. 4), pp. 42-49.

16. Stegmann, R., Koch, M., Lacher, M., Leckner, T., and Renneberg, V. (2003). Generating Personalized Recommendations in a Model-Based Product Configurator System, In <u>Proc. of Workshop on Configuration 2003</u> held in conjunction with 18th International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico.

17. Wind, Y., Mahajan, V., and Gunther, E. (2002). <u>Convergence Marketing – Strategies for Reaching the New Hybrid Consumer</u>, Prentice Hall, USA.

Using knowledge-based advisor technology for improved customer satisfaction in the shoe industry

Werner Falle, Daniel Stöfler, Gallus Herrenschuh GmbH, Lagerstrasse 4, 9400 Wolfsberg
werner.falle@gallus.com, daniel.stoefler@gallus.com

Christian Russ, ConfigWorks GmbH, Universitätsstr. 94/7, 9020 Klagenfurt
cruss@configworks.com

Markus Zanker, Alexander Felfernig, Universität Klagenfurt IWAS, Universitätsstr. 65-67, 9020 Klagenfurt
markus@ifit.uni-klu.ac.at, alf@ifit.uni-klu.ac.at

ABSTRACT

Due to structural changes European shoe manufacturing has become very challenging over the last years. These changes lead to increased customer orientation but also require differentiation of products and processes. Not adaption to changing environments, but control of change and designing the future are required in order to ensure sustainability for shoe manufacturing in Europe. The *MyGallus* project follows this approach and combines the strategies of Mass Customization and intelligent IT-systems in order to change classic sales and production business logics.
In this paper we present a way how a knowledge based advisor system with underlying product configuration technology can be implemented in order to support a new form of customer care and product recommendation over virtual sales channels. Business processes develop themselves to a truly customer oriented business-logic by allowing the customer to individualize *her/his* product according to *her/his* very personal requirements. Intelligent advisor systems can help the customer in the product selection process. Additionally, there is a potential for sales process automation and cost optimisation for the company itself.

KEYWORDS

Shoe industry, Mass Customization, product configuration, knowledge-based advisor systems

1. INTRODUCTION

Customer orientation is vital to the success of the European Shoe Industry in facing structural changes, "cost-focus" in pricing strategies and global competition from low-wage countries: Understanding customers and their requirements, creating a relationship with them and integrating them to the company's supply chain is essential for the sustainability of the entire industry and can only be achieved by the creation and implementation of innovative IT-systems [EuroShoe Project (2002)] [Commission of European Communities (2003)] [Commission of European Communities (2001)]. The submitted *MyGallus* project[18] is taking this challenging approach into account and combines strategies of knowledge based advisor systems and configuration technology in order to find a new way of virtual customer service and customer care. The concepts of advisor systems will help to identify the specific customer requirements and

needs, whereas the underlying configurator system will guarantee a valid and technical possible solution of the matching shoe. The development and design of such "customization systems" is a precondition for what could lead to a major change in the shoe industry`s business logics, namely moving from a "product-push" to a "customer-pull" oriented value chain. Offering personalized products at nearly the same price and with short delivery time is a major challenge in management that effects production and distribution. Configuration systems, their effectiveness and finally their acceptance from the point of view of the customers are the important topics.

We will describe the implementation of a hybrid advisor system with an underlying product configurator and the embedding into existing IT-infrastructure and business processes. Finding a model with an intelligent mapping mechanism of the customer situation onto the product world to find his/her product is the starting point for applying Mass Customization concepts in the shoe industry.

Therefore the paper is organized as follows: in Chapter 2 we present the research project *MyGallus* and discuss the challenges in shoe industry. Starting with the investigation of the customer`s behavior and target groups we discuss the customer`s life cycle and derivate objectives and targets of the project itself. In the third chapter we present how a hybrid knowledge based advisor system can be used and integrated into the existing IT-infrastructure to offer the customer better fitting shoes with the added value of individualization and personalization of the recommendation and selling process of the product itself. In detail the combination of advisor systems and constraint satisfaction based on configurator technology is discussed and a integrated solution is figured out. Finally, we give conclusions and an outlook on future work.

2. PROJECT OVERVIEW

2.1 Market challenges and consumer behaviour

The international shoe industry is changing its macroeconomic structure, moving from a sellers' market to a saturated buyers' market: These markets are characterized by a strong customer focus, requesting added value of the offered products and leading to a high degree of product differentiation [Brenton, P. et.al. (2000)].

Meeting customers' requirements becomes essential for a company's success and means offering a wide range of products to satisfy highly varying customer needs in order to acquire and maintain an USP. Still substitutes are easily available and markets are characterized by tough competition and aggressive pricing strategies, further accelerating companies' ongoing efforts of minimizing costs.

Common mass production philosophies are unable to cover those requirements of adding value through differentation: fast reaction within the production process, easy switching within a huge variety of products and minimizing time-to-customer

and costs at the same time are the challenges that can be faced only by innovative business models. New manufacturing structures were needed and have been found in team work concepts and agile manufacturing approaches that are able to face the challenges of reacting fast and efficiently to those requirements [Blecker et.al. (2004)].

Nevertheless the industry is still relying on its traditional way of doing business by offering products that are pushed on the market and distributed by standard marketing and sales activities. However, adaption to changing environments is not enough; action rather than reaction is required in order to survive.

2.2 Customer Life Cycle

Implementing a Mass Customization philosphy can be seen as a customization process by itself: By changing existing business logics, mass customization concepts require the definition of a "fitting" solution space for a specific company, offering its products to its customers on its markets. Most critical to the success of the project are therefore Gallus' customers, their preferences and their behaviour: It is necessary to take care of all touch points that occur within the customer life cycle. These touch points can be separated into four categories/patterns: Engage, Transact, Fulfill and Service (ETFS). Customer orientation means understanding these touch points in a specific context and to respect them throughout all phases of customer interaction, not only concentrating on the sale of men shoes itself [Meta (2000)].

Firstly, a clear and effective communication of advantages and added values is essential to let the customer know about possible benefits of the individual product. Getting a customer's attention and interest is a precondition to further involve him/her into configuration, leading to "give away" personal preferences during the customization process of his/her personal shoe [Imhoff et al. (2001)].

After attracting the customer to the product by pointing out its (possible) features and its degree of personalization, he/she needs to be guided and serviced through the selling process. This can be achieved by advisory systems, that will be used for creating customer satisfaction and establishing a relationship on virtual sales channels. However, customer care continues after purchasing: Using, evaluating and integrating all data gained during the customization process by advisory systems will allow to get loyal customers, whose satisfaction with the product or service urges other customers to join the cycle and that will make them moving through the cycle again and again.

Therefore, throughout the entire project the ETFS customer life cycle model is used as a guideline to fit all components of the marketing mix to the specific definition of the *MyGallus* solution space and to grant a truly customer oriented business process.
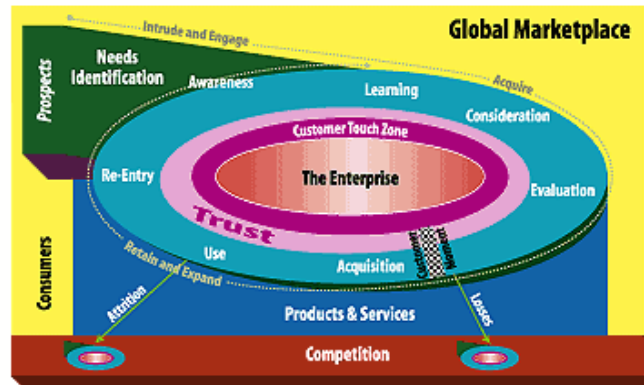
Figure 20: Customer Life Cycle [Imhoff et al. (2001)]

The advisory system is one part of the overall project to enable and guarantee the customer contact and guidance of the product selection process as well as a mechanism to learn about the customers' preferences and needs. This again can give the company valuable knowledge about the customers, how they think and what they really want. New ideas and product innovation can be gained by analysis of the interaction history of the advisory system.

2.3 Project objectives

Based on the before mentioned arguments the project's main target is to ensure a sustainable shoe production implementing strategies of Mass Customization. According to [Hammerle et.al. (2003)] the project is oriented towards customers and employees needs as well as towards a sustainable use of resources. As can be seen from Figure 21, there is a closed loop from the customer and requirements elicitation, to individualized products, new production concepts until distribution.

During the project duration it will encompass the following steps in detail:

- ✓ Evaluate the market situation and customers needs as well as possible customer groups
- ✓ Find ways to individualize and personalize shoe products without cost increase
- ✓ Development of appropriate IT-support for improved advise and selling at the point of sale
- ✓ Adapt the product management to cope with the new situation
- ✓ Introduce new flexible production concepts and team work mechanisms
- ✓ Explore new distribution channels and strategies to handle the customer oriented selling

Figure 21: The interactive advisor system as a central communication link between company and customer

3. A HYBRID KNOWLEDGE BASED ADVISOR SYSTEM

A customer individual shoe is a combination of a selection - and customization process. Each person has an individual foot with special requirements, so at the first step a suitable and satisfying basic shoe form is necessary. In the second step the customization of the basic shoe type can take place. There the customer can choose between different options for the shoe like color, leather type, size, laces, sole-type, shoe lifts etc.

The presented approach differs in several ways from the class of successful configuration systems that base their proposals on a product centric search of valid product attribute combinations that meet the users' requirements [Bergmann R., Traphoner R., Schmitt R., Cunningham P., Smyth B. (2003)] [Bridge D. (2001)]. This systems are performed best in very technical domains like telecommunication switches or IP-VPNs [Ardissono L. et.al. (2002)] [Ardissono L. et.al. (2003)] where deep knowledge about the technical product domain is necessary[19]. Our idea, with a knowledge based advisor system, follows a hybrid approach [Burke R. (2002)] and makes extensive use of explicit sales force - and expert knowledge of the shoe domain. Based on a conceptual model for the advisory process we design and realize a knowledge based advisor system that identifies the specific customer needs and calculates recommendations that satisfy the companies' internal restrictions as well as the customers' requirements. [Ansari A., Essegaier S., Kohli R.(2000)] [Blecker et.al. (2004)].

Figure 22 gives an overview on the major concepts of our advisory knowledge base. The advisor system consists of a static model with the customer properties and the product properties and a dynamic model with the business rules for the result reasoning process, the personalization and interaction behavior.

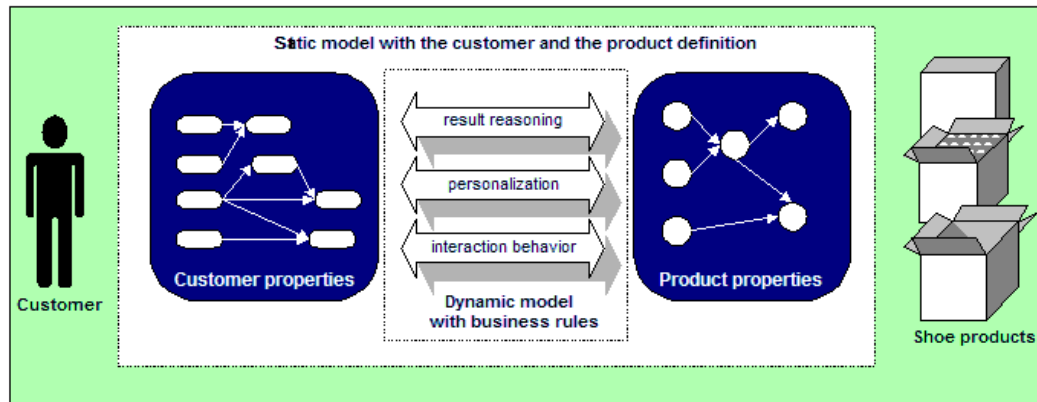[19] EU research project CAWICOMS www.cawicoms.org

Figure 22: Conceptual model of the declarative advisor system knowledge base

The *customer properties* are the variables in the advisory process that represent the current user's[20] profile with the special interest and needs. The corresponding values are determined by questioning directly the user or by incorporating existing information over the user from external resources like customer databases, registrations forms etc.

The *product properties* specify and describe a meta model of the product structure and attributes. These product characteristics are essential for the reasoning and search process of valid and satisfying recommendations for the user. Note that the concrete instances of products can be imported from existing legacy systems like a product catalog or an ERP[21] system.

The *business rules* are the connection between the users- and the product world. All of this knowledge can be expressed in a high-level language in form of filter rules, constraints, and personalization rules. The selection of a suitable product is based on the filter rules relating customer properties with matching product characteristics [Bridge D. (2001)]. On the left hand side a filter rule consists of an expression on arbitrary customer properties describing conditions when the filter has to be applied. On the right hand side the technical restriction on the product properties is defined. The business rules language comprises relational, arithmetic and logical operators between variables and values. Additionally an intelligent priority mechanism helps the reasoning engine to relax applied filter rules if there are no results in the search process left. This kind of "soft-rules" enables the system to behave like a sales person to present alternatives if certain user requirements can not be satisfied.

Typical forms of business rules are:
- Filters: "If the customer is of business type and heavily uses the shoe, recommend products of business type with high quality"
- Constraints: "If the customer wants a brand shoe with a low price, inform him or her that he might have to relax his restriction on the price"

---

[20] We use the term user and customer synonymous for an entity requesting shoe products.
[21] Enterprise Resource Planning

- Personalization: "If the customer is a young person, ask him for casual shoes first and give him a hint on the low cost brand"

We have chosen an "if-then" style representation which is understandable for non-technicians i.e. the domain experts.

For the personalized ordering and ranking of product recommendations, we used a utility – based approach relaying on MAUT (Multi-Attribute Utility Theory) [Ardissono L. et.al. (2002)] [von Winterfeldt D., Edwards W.(1986)]. Therefore we introduce high-level interest dimensions like "economy", "fashioness", "prestige" between the customer and product properties and define corresponding utility functions. During the advisory process, the engine calculates an estimate of the customers' relative interest in these dimensions by evaluating the answers of the customer properties. Given this information, the engine computes an overall utility and a corresponding range of each matching product.

For the configuration mechanism we exploited a CSP (Constraint Satisfaction Problem) based solver [Tsang, E. (1993)] where we map the customer properties of the advisor domain to the variables of the constraints solver. In CSP there are variables with a defined domain, which can have a finite enumeration or a defined domain like real numbers, whereby each variable can also have multiple values. Additionally there are constraints among these variables which restrict the variables to specific domains. The user of a CSP can define its requirements by defining an initial set of values. The solution to a CSP is to find a value assignment s.t. each variable holds a value from its domain and all constraints and customer restrictions are satisfied.

The mapping between the advisor engine and the CSP solver (Figure 23) is done by masking areas of the customer properties with the variables of the CSP problem and to interchange values between them. So, for directly acquired customer properties, each variable corresponds to a question and the allowed domains represent the possible answers of the question. If the advisory process is finished and the satisfying products are found, the configuration process can start by solving valid value combinations for the open variables without violating the constraints. Finally the customer would get a list of recommended shoe products which fit to the basic requirements like sole type, leather type und shape with the individualized color and laces, shoe lifts and a price the customer accepts [Friedrich G., Stumptner M. (1999)] [Fleischanderl G. et. al. (2001)].
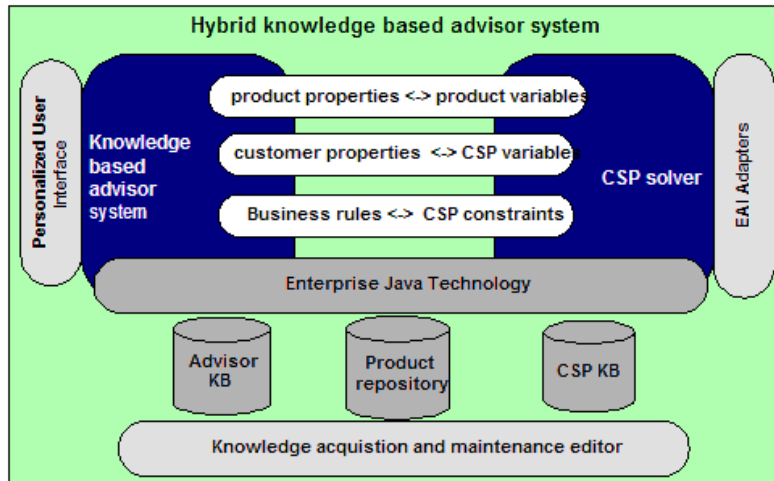
Figure 23: Architecture of the hybrid knowledge based advisor system with CSP integration

It is very important to note that the whole knowledge base of the advisor system can be maintained by a graphical editor (shown in Figure 23 as knowledge acquisition and maintenance editor) implemented in Java Swing Technology. This platform independent Windows GUI guarantees an easy to use and powerful editor for the creation and change of the domain knowledge and the behavior of the interactive advisor system. Based on our experience in the CAWICOMS project we will supply templating mechanisms to generate automatically adaptive Web-based user interfaces for the customer advisory process. This enables the company itself to easily change and modify the behavior and the appearance of the advisory system and ensure an effective and up to date customer guidance and recommendation of the product portfolio [Gurzki T., Schweizer P., Eberhardt C.-T. (2003)].

The technical implementation framework is based on Enterprise Java Technology[22], where API like Enterprise Java Beans, JDBC and the Java Enterprise Application Integration Adaptors (EAI) are used to realize the overall functionality for the knowledge based advisor system. This includes a multi user server system which can handle multiple advisor sessions, establish database connections and a Web-based user interface for the frontend representation of the advisor system. These well established programming technologies guarantee a robust and reliable environment for fast implementation, which is well known by developers and can be easily learned and extended by external solution providers.

To follow a simple example we implemented a horizontal prototype where a simplified advisory process is visualized graphically (Figure 24). First the customer would be welcome and informed by Mr. Shoe (the virtual selling assistant) what he or she can do here. Then the elicitation phase would start, where the customer is asked for her/his special needs, preferences and restrictions on a good shoe. During this phase it could happen that the customer has incompatible requirements, so Mr. Shoe would show the conflict and will help to remove the unsuitable combination. The next step will be the recommendation of a list of valid shoes by Mr. Shoe with the predefined

list of configurable properties like the color, sole type, laces etc. The customer is now able to choose a shoe from the list and to configure interactively the final satisfying product. Note that for the customer the user interface of Mr. Shoe looks like a virtual assistant who helps him or her to find and configure the best fitting shoe. In the background an advisor system and a CSP solver interoperate to support the customer in the search process without a change in the behavior or appearance. So the advantages of a user centric advisory dialog and recommendation with an additional technical configuration can take place in one integrated system. The customer gets a better guidance and assistance during the search process for a shoe and also gets intelligent explanation support why these recommendations are the best solution for the current situation.
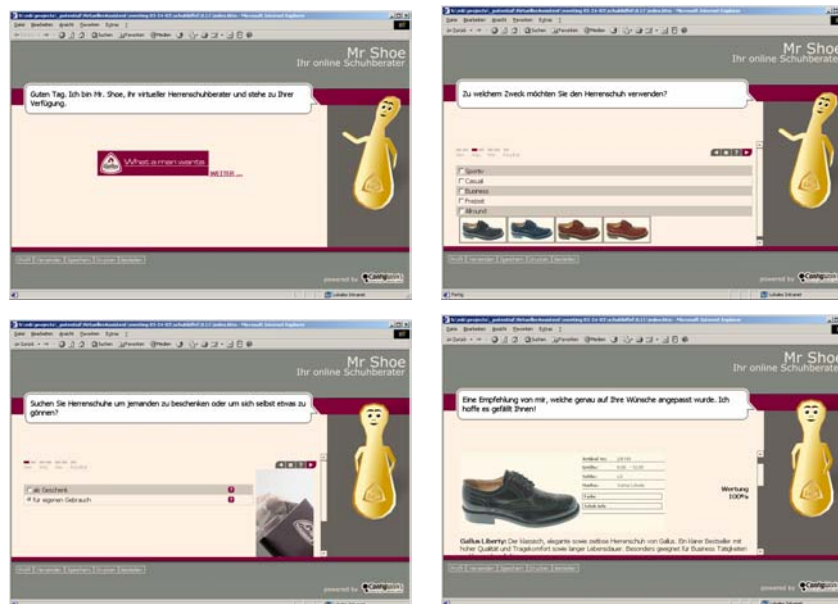

Figure 24: Simple prototype scenario of the interactive shoe advisor Mr. Shoe

4. INTEGRATION INTO THE EXISTING IT-INFRASTRUCTURE

In the *MyGallus* project the realization of the advisor system is the attempt to use interactivity and guidance to improve the customer satisfaction during the shoe searching and selling process. Consequently the overall integration of the advisor system into the existing IT-infrastructure and the adapted business processes is essential for the success of the whole business model. From the company view, there are existing IT-systems which support the internal information management and the production process. These data and processes are very product oriented and specialized on the optimization of time, resources and cost. On the other side there is the customer, who is not interested in the specialties of companies' internals like technical product details and manufacturing processes. Moreover the customer has an individual situation with needs and wishes. She or he wants to get an optimal proposal her/his new pair of shoes. So there is a knowledge-gap between the abilities and competences of the

---

company and the customer specific needs. Companies must learn to address the customers' needs instead of presenting technical products features and parameters [Pine II et.al. (1993)].

As presented in the figure above (Figure 25) the advisor system aims as an intelligent mediator between the customer and the company view of their interest at the virtual channel. The advisor system tries to communicate with the customer and to transform the customers' requirements to the technical product world to find a matching solution. Therefore interfaces to the customer and an exchange interface to the ERP-System as middleware are needed. From the customer side this is done by the adaptive user interface (frontend) which offers personalization features to simulate the intelligent selling assistant Mr. Shoe [Ardissono L. et. al. (2002a)]. The connection to the available products and variants is done by an integration layer with XML technology. Data like shoe models and attributes of the models, configuration information, as well as the amount and the availability need to be exchanged so that the advisor system has an appropriate product basis for customer guidance. This is very important, because the advisor system may only recommend and offer shoe models which are currently available in the production program or on stock. Depending on the season and the production output the models can vary and this must be synchronized immediately with the advisor system itself.

If a customer accesses the homepage or a self-service terminal in the shop and he/she will have the possibility to use the help of the virtual assistant Mr. Shoe. Therefore he/she answers the offered questions and navigates through the advisory process. At the end the virtual assistant will calculate a satisfying solution which matches the customers' needs as well as archives the technical restrictions of the product parameters. Additionally the customer will get an explanation with arguments why the suggested products are the right ones. The customer could order the shoes or get in contact with the nearest merchant to try the product.
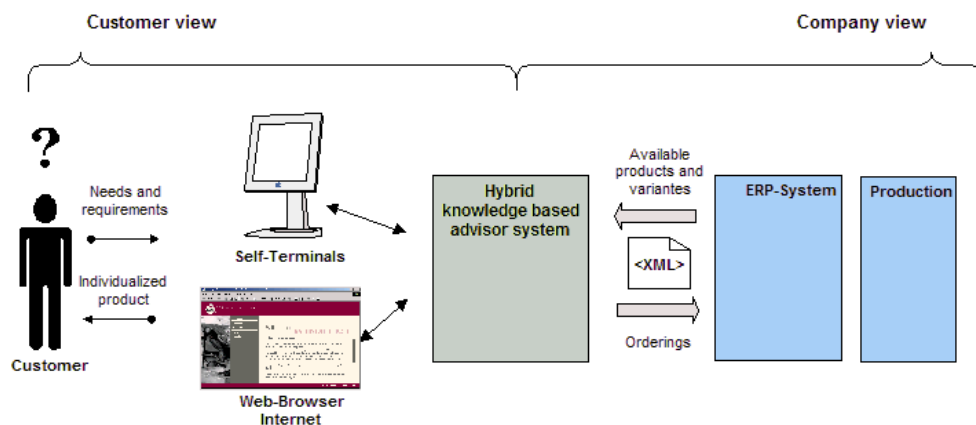

Figure 25: Integrated view on the hybrid knowledge based advisor system

For the company, the advisor system will enable new potentials because of the intermediate and realtime information of the customer advisory sessions and the seamless integration into the existing ERP-System. So the sales

department will be able to maintain and manage the advisory process by themselves, and can change the sales strategy and occurrence instantly. This will make them more flexible and guarantee a unique presentation and communication of the product information to the customer. In addition the sales persons can be trained themselves by the advisory system to get the right questions to the customer, present the newest shoe products available, as well as learn the selling arguments why the shoes are the best recommendation for the customer.

The marketing department is able to analyze the customer advisory sessions and develop new marketing strategies and can provide the product management with useful information of the customers' preferences and requirements. With the customer information the product manager will be able to design more suitable products and features with less efforts and expense. These newly created products will enter in the production and selling process and can be offered by the virtual assistant Mr. Shoe in the future. So the information lifecycle between the requirements of the customer and the products of the company is closed and a continuous loop of improvement can be facilitated.

5. CONCLUSIONS AND FURTHER WORK

In this paper we have shown the attempt to find new ways and strategies to make the Austrian shoe industry more competitive by offering improved customer orientation and flexible sales models. Knowledge based advisory is a promising approach for companies to add value to its products for the customers on different and innovative sales channels like the Web and self terminals in shoe stores. We have presented a hybrid approach that combines several techniques for customer guidance, product selection, personalized ranking and product individualization with configuration functionality. The knowledge acquisition and maintenance problem for the up to date information of the products is addressed by providing adequate editors and the easy to use generation mechanism of the customer Web-based user interfaces of the advisory system [Smyth B., McGinty L. (2003)].

Our future work will concentrate on the enhancement of the advisory system regarding special needs in the shoe industry. Especially the handling of the physical distance between the customized shoe and the customer during the recommendation process; the customer can not feel, touch and try the shoe on; this has to be solved by reducing the psychological distance via the virtual channel [Gilmore, J. H., Pine, B. J.(2000)]. Additionally we will think about improved mechanisms for online and offline analysis of the advisory sessions: Why did users cancel a session, how could recommendation dialogues be optimized and shortened without losing quality in the final solution are issues that need to be discussed. Data Mining as an additional opportunity to learn more about the customer needs to be integrated into further research as well, especially thinking of open innovation learning possibilities. Further work will also be done on the organizational and technical integration of the advisory system into the existing sales channels and the IT-infrastructure. So we learned that a virtual sales assistant is often not accepted by the sales staff because of rivalry and competence reasons. The sales staff has objections and concerns against the new

"sales member" and does not accept the use of the system. Here adequate education and training of the staff has to be considered and the benefit of the use of the advisory system has to be promoted to the customers as well as to the company staff itself.

Technically, the smooth and stepwise integration of the advisory system into the existing infrastructure is crucial for to the success of the overall project. Besides the technical data exchange, the integrated system must support the different business processes in production, selling, billing and distribution. The performance of the overall system and the real time behavior of the advisor system during high traffic access has to be simulated so that stability and uptime can be guaranteed.

Reduction of cost and time to customer as well as the improvement of efficiency can only be achieved if we can manage the whole customer life cycle in terms of supporting the entire searching-, deciding- and buying process of the customer.

The result of the project will be a conceptual model and a realized prototype of an intelligent shoe advisor, working on a hybrid knowledge based advisory system that integrates concepts of the IT-domain with sales strategies and marketing. The outcome is a reference model of a new way to improve customer relationship by establishing and optimizing contact and communication with interested customers groups. Besides that, with the physical output of the project, the personalized shoe, this project should also be a "living proof" of a useful and effective Mass Customization application in the shoe industry and a first step in achieving the vision that "there will not be any standardized shoe produced in Europe by the year 2010".

ACKNOWELEDGEMENT

REFERENCES

Ardissono L. et. al. (2002a), *Customizing the Interaction with the User in On-Line Configuration Systems,* Proceedings 15th European Conference on Artificial Intelligence - Configuration Workshop, Lyon, France.

Ardissono L. et.al. (2003), *A Framework for the Development of Personalized, Distributed Web-Based Configuration Systems*, AI Magazine, Vol. 24(3): Fall 2003, 93-110, AAAI Press.

Ardissono L. et.al. (2002), *A Framework for Rapid Development of Advanced Web-based Configurator Applications*, Proc. 15th European Conference on Artificial Intelligence - Prestigious Applications of Intelligent Systems (PAIS), Lyon, France.

Ansari A., Essegaier S., Kohli R. (2000), *Internet recommendation systems*, in: Journal of Marketing Research, 37. Jg. (2000), H. 3 (Aug), S. 363-375.

Bergmann R., Traphoner R., Schmitt R., Cunningham P., Smyth B. (2003), *Knowledge-intensive product search and customisation in electronic commerce*, in: E-Business Applications. Springer Verlag, 2003.

Blecker et.al. (2004), *An Advisory System for Customers' Objective Needs Elicitation in Mass Customization*, Proceedings of the 4th Workshop on Information Systems for Mass Customization (ISMC 2004) at the fourth International ICSC Symposium on Engineering of Intelligent Systems , University of Madeira, Funchal/Portugal.

Brenton, P. et.al. (2000), *Adjustment to Globalisation: A Study of the Footwear Industry in Europe*, The Centre for European Policy Studie (CEPS) - Thinking Ahead for Europe, http://www.ceps.be/Pubs/2000/wd/151.php

Bridge D. (2001), *Product recommendation systems: A new direction*. In R.Weber and C. Wangenheim, editors, Procs. of the Workshop Programme at the Fourth International Conference on Case-Based Reasoning, pp. 79-86.

Burke R. (2002), *Hybrid recommender systems: Survey and experiments*, User Modeling and User-Adapted Interaction, (4):331-370.

Commission of European Communities (2003), "*Economic and* Competitiveness Analysis of the European Textile and Clothing Sector", http://europa.eu.int/comm/enterprise/textile

Commission of European Communities (2001), *Report "Textile and clothing industry in the EU: A survey"*, Enterprise Paper No 2-2001, http://europa.eu.int/comm/enterprise/textile

EuroShoe Project (2002), *The Market for Customized Footwear in Europe Market Demand and Consumers' Preferences*, EuroShoe Project Report http://www.euro-shoe.net/

Fleischanderl G. et. al. (2001), *Constraints as key to industry configurator solutions* , Elektrotechnik und Informationstechnik (e&i), 7/8/2001, 349 – 354.

Friedrich G., Stumptner M. (1999), *Consistency-Based Configuration,* AAAI Workshop on Configuration, pp 35-40, Orlando Florida.

Gilmore, J. H., Pine, B. J. (2000), *Customization that counts*, in: James H. Gilmore / B. Joseph Pine (Hg.): Markets of one: creating customer-unique value through mass customization, Boston: Harvard Business School Press 2000, S. vii-xxv.

Gurzki T., Schweizer P., Eberhardt C.-T. (2003), *A virtual-sales-assitant architecture for e-business environments*. In E-Business Applications, pp. 77-86. Springer Verlag.

Hammerle et.al. (2003), *Nachhaltige Produkte und Dienstleistungen – Leitfaden zur Entwicklung zukunftsfähiger Geschäftsfelder*, Berichte aus Energie- und Umweltforschung 10/2003, BM f. Verkehr, Innovation und Technologie.

Imhoff et al. (2001), *Building the Customer-Centric Enterprise*, John Wiley & Sons, http://www.dmreview.com/article_sub.cfm?articleId=2813

Meta (2000), *Engage – Transact – Fulfill – Suppor Modelt*, Meta Group Inc., Stamford, CT , www.metagroup.com

Pine II et.al. (1993), *Making Mass Customization Work*, Harvard Business Review, Harvard.

Piller, F. and Sabol, T. (2003), *Mass Customization und Kundenintegration*, Symbosium, Duesseldorf.

B. Smyth and L. McGinty. *The power of suggestion*. In 18th International Joint Conference on Artificial Intelligence, pp. 127-132, Acapulco, Mexico, August 2003. Morgan Kauffman.

Tsang, E. (1993) *Foundations of Constraint Satisfaction*, Academic Press, New York.

Tseng, M. and Piller, F. (2003), The Customer Centric Enterprise, Springer, Duesseldorf.

von Winterfeldt D., Edwards W.(1986), *Decision Analysis and Behavioral Research*, Cambridge University Press, Cambridge, UK.

THE ORDER SPECIFICATION DECOUPLING LINE (OSDL)

Benjamin Loer Hansen, Lars Hvam

Department of Manufacturing Engineering and Management
Technical University of Denmark, Building 423, Lyngby 2800
Email: blh@ipl.dtu.dk

This paper discusses the important issue in build-to-order production of separating innovative long term engineering processes from day-to-day variant specification processes, i.e. the processes creating specifications such as bill-of-materials, drawings, text descriptions, routings etc. The paper proposes and discusses the term Order Specification Decoupling Line (OSDL), which is to be used to explain the separation of industrial specifications into some information created prior to order acquisition and some information created during order acquisition and order fulfillment. For instance one may find some welding specifications or product descriptions which are standard for every order, while there may be other specifications, such as bill-of-materials and drawings, which are customized. Through a use of this concept it becomes possible to discuss different levels of OSDL and different variant specification tasks, such as engineer-to-order, modify-to-order, configure-to-order and select to order. It must be emphasized that the paper is more of a research proposal for discussion, than actual conclusive findings.

1. INTRODUCTION

This paper is focused on operations management in so-called build-to-order (BTO) companies. Examples of such companies are manufacturers of custom engineered air-conditioning equipment, doors and windows, wind mills, industrial boilers, steel chimneys etc.

Selling, manufacturing and delivering such customized products often result in complex operative processes. When selling and manufacturing e.g. customized windows the company may in principle not know *how much* and *what* is to be produced next week, and *how* it can best be produced and delivered.

One task of the operations management in BTO companies is related to the cross functional "paper-work" processes needed during order acquisition and order fulfillment before any physical production can be started. This is typically the subtasks of specifying customer requirements and product features (e.g. a specification sheet or a quote), customizing products (e.g. drawings), evaluating manufacturing methods (e.g. routings), delivering the final product directly to the customer (e.g. ship-to and bill-to addresses), etc. The output of this kind of processes is information typically in the form of documents, such as bill-of-materials, drawings, routings, quotes, diagrams etc, i.e. the output is information. In the present paper this information is called specifications and the processes are called *variant specification processes*
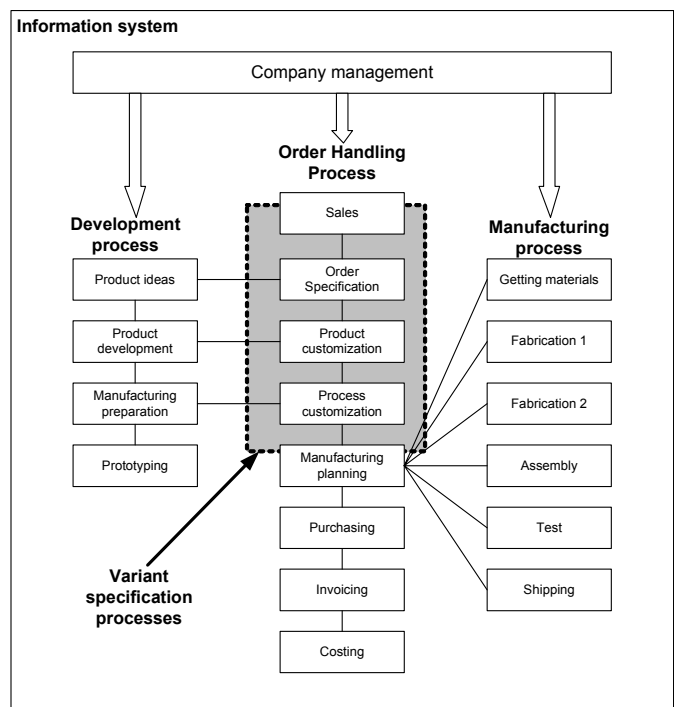

Figure 1: Development and variant specification processes in the Activity Chain Model.

[51], [52]. In BTO companies these processes basically creates variants of specifications, based on existing knowledge and information. Terms like custom engineering, build-to-order [53], engineer-to-order, configure-to-order and mass customization [54] are all related to this subject. In this paper BTO is perceived as the overall manufacturing concept covering these terms.

The target of the present paper is these variant specification processes in BTO companies. Mass production of standard products on one hand and development of innovative products (e.g. new medicine) on the other is not covered by this paper. The question raised and discussed is:

- What does it imply for BTO companies to separate day-to-day variant specification tasks (custom-engineering, configuration etc.) from long term development efforts, and how can the concept of an order specification decoupling line can be used to describe this problem?

2. SEPERATING PRODUCT DEVELOPMENT
FROM DAY-TO-DAY CUSTOMISATION

In most BTO companies it is possible to identify a group of long term development activities targeting an entire market, and a set of day-to-day activities serving individual customers. However, a widely observed problem is that day-to-day operations consume all resources resulting in poor long term development. In order to achieve effective long term development as well as effective day-to-day variant specification processes it is important to seek to separate development from daily operations as illustrated in figure 1. The focus should be on processes and on satisfying the customers of each process, facts that have been discussed for years in the Business Process Reengineering literature [55] for years (and targeted specifically in the "Activity Chain Model" [56] illustrated in figure 1). The process focus is one of the main reasons for focusing on the subject *variant specification processes*.

Therefore it is a task of strategic importance in BTO companies to define the allowed product variance and prepare knowledge and information bases to support efficient variant specification processes.

One solution scenario might be to prepare (develop) relatively few fixed rules regarding product platforms and generic product structures, and few restrictions regarding the usage of parts and materials. This scenario makes it possible to engineer a wide variety of highly customized products, but the down-side is long and costly variant specification processes (engineering) and a subsequent slow and expensive non-standardized production.

Another solution scenario is to predefine generic product structures, standard modules and parts, and a set of well defined rules and calculations for configuration. This scenario makes it possible for engineers to configure customized products fast, and in a good quality, without consuming too many resources. The down-side in this scenario is large investment in the development of modules, product structures and rules, and a limitation with regards to the degree to which products can be customized. This may prove fatal if the development efforts miss actual customer desires and sales fail.

These scenarios illustrate an important strategic question facing a BTO company: what product and production information should be developed and defined before any products are sold, and what should be specified during actual order acquisitions and order fulfillment processes?

3. ORDER SPECIFICATION DECOUPLING LINE

When it comes to manufacturing planning an important concept is the "Customer Order Decoupling Point" (CODP). This concept indicates a separation of manufacturing processes in those producing to stock and those producing to order: *"the CODP is the point in a value-adding process where a customer penetrates the operative system and where a product is linked to a specific customer order. Downstream from this point, operations are order driven; upstream from this point, forecast driven"* [57]. Another term for this concept is the "pull vs. push line" [58]. The CODP or pull/push line is traditionally focused in material planning and positions of material stocks.

The central hypothesis of the present paper is that in relation to variant specification processes it is relevant to focus specifically on information flow. Similar to material flows it is relevant to decide on what extent of specifications, i.e. product and production related information that should be defined prior to order intake, and what should be created during order handling, i.e. variant specification processes. Therefore the proposition of the present paper is to extend the concept of the Customer Order Decoupling Point to information flow and introduce the term Order Specification Decoupling Line (OSDL). This term covers the fundamental separation between "development" and "variant specification": *The OSDL is the fundamental "line" that separates the pre-developed specifications (including predefined standard elements in custom made specifications) from those specifications made during variant specification processes (order acquisition or during*

order fulfillment), i.e. separation of information defined to general markets and information directly created in relation to individual customers.

The information made explicit prior to order acquisition typically consists of material, part, module and product specifications defining structural "building blocks", and specifications of manufacturing processes, generic routings, generic test procedures etc. Rules, constraints and calculation methods can also be made explicit and defined prior to an order and may thus be said to be part of the information defined prior to an order. Additionally there can be predefined document templates for how to make the different documents/specifications, for instance headers and footers and standard text descriptions.

The information made explicit in the order related variant specification processes is thus partly combinations and modifications of predefined information, partly new information added during the processes. The complexity of creating these specifications depends on the amount of information that has been predefined, and thus the OSDL. These issues are illustrated in figure 2 and 3. Moving the OSDL from the left side towards the right in figure 2 and 3 is thus done through an explicit specification of standard parts, standard subassemblies and modules, standard geometries, predefined rules, constraints, document templates etc.
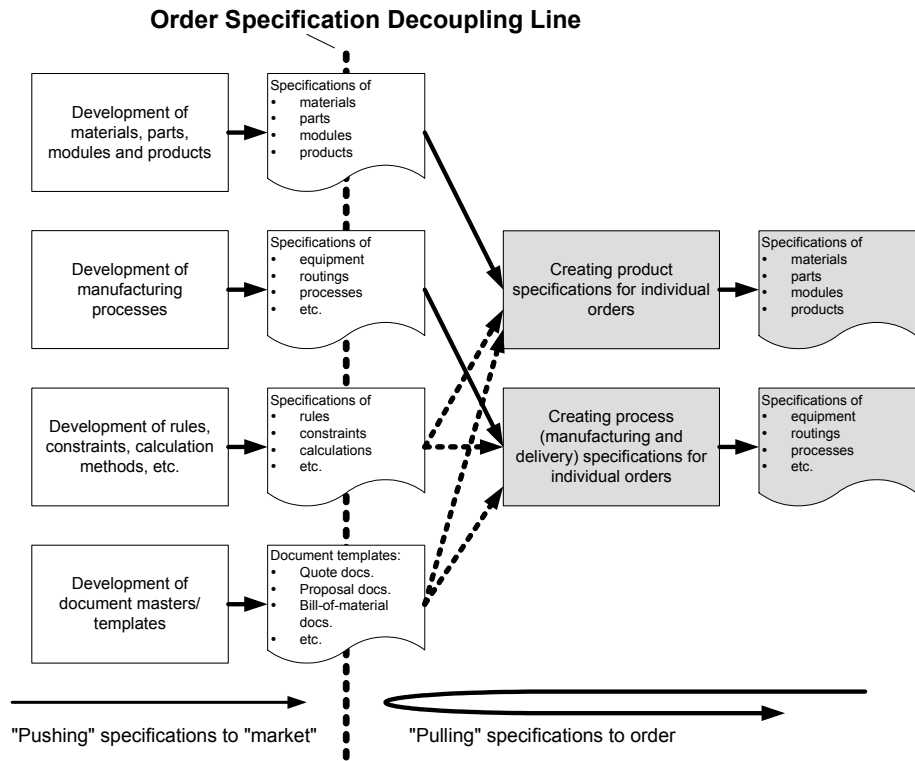

Figure 2: Illustration of the order specification decoupling line. Specifications market with white are pushed to market. Grey specifications are pulled to order.

The OSDL is highly related to the CODP, but somewhat different, which can be illustrated by the case of producing standard products. Here the OSDL is fixed at predefined standard product structures and manufacturing methods (i.e. standardized specifications), whereas the CODP in the same case can be positioned both on the level of finished products, on the level of assemble-to-order and on the level of producing even subassemblies to order.

It must be emphasized that the concept of extending the CODP concept with a more information focused OSDL is more of a research proposition and a topic to support discussions, than an actual conclusion.

4. FOUR GENERIC LEVELS OF OSDL

The concept of the OSDL can be used to define different classes of variant specification processes as illustrated in figure 3 (these are closely related to similar levels of the CODP found in the operations management literature [59], [60]. At least four generic types can be identified:

- Engineer-to-order process
- Modify-to-order process
- Configure-to-order process
- Select-to-order process

These four types are examples of classifications for which other names are often used. Real processes fill a continuous specter of which many are in between the proposed types. In the following, there is a short description of the different types.

Engineer-to-order processes are typically taking place in systems consisting of skilled technicians/engineers and support tools such as CAD/CAM, simulation tools and various data storages. This kind of processes are characterized by complex specification activities and a relatively little degree of pre-developed product parts and modules. It can be discussed whether this type creates variants of specifications or whether it creates totally new information and knowledge (i.e. not variants). Often this will depend on the actual customer related processes. The pre-developed specification foundation typically consists of norms and standards, which together with material specifications are used as a base for engineering new products. Characteristic for the processes is thus the creation of new parts and new part numbers, which cannot be predefined in for instance ERP systems, and for which there may be no specific predefined production method, neither pre-calculated costs. This type of processes has been identified as typical for products such as cement plants, milk powder plants, industrial boilers, and wind mills [61], as illustrated in figure 3.

Modify-to-order processes are somewhat similar to engineer-to-order processes. However, the OSDL is moved towards the customer, so that less creative engineering skills are needed. Often the specification tasks are based on well-defined patterns of product structures, where the allowed variety is less than found in traditional engineering. A typical solution is to use a base of generic modules and product patterns, which can be "cut-to-fit", during variant specification processes. Tailor made clothes and custom-made windows are made in this way: Standard patterns exist, from which the final modules and products can be made. Thus, the variant specification processes can define customised products faster and cheaper. The systems in which these processes take place are similar to engineer-to-order systems, i.e. skilled people and support tools such as CAD/CAM and simulation tools. Often one will find a high degree of parametric 3D models used as a basis

Figure 3: Illustration of different levels of the Order Specification Decoupling Line. Grey arrows indicate generic types of processes which create specifications to order. On the right side is given examples [61] of Danish products which are customised to different degrees.

for creating new product variants quickly. Examples of products related to this kind of processes are industrial silencers, industrial chimneys, and industrial gates [61]. All these products are cut out of metal plates in generic patterns, but with different dimensions.

Configure-to-order processes often take place in relatively automated systems, typically in the form of expert systems/configurators or engineers using special applications to support the generation of specifications. In configure-to-order processes, the specification task is simplified to a question of how to combine existing product parts and modules. Together with such parts and modules, rules and constraints of valid combinations will typically also have been developed. Thus the OSDL is moved even more towards the customer, through large development efforts. With a large degree of specifications made prior to the order, this kind of specification processes are characterized by speed, low cost and few errors, making it suitable for mass customisation. Some examples of products which have been observed in this group in the Danish industry are air conditioning equipment, electric switchboards, pumps and hi-fi equipment [61].

Select-to-order processes create only few new variants of specifications, i.e. product specifications on invoices with prices and delivery details. Thus the task is that of transforming information of customer needs to a specification of a standard product, -and retrieving the corresponding specifications for manufacturing from a data storage. The OSDL is moved all the way towards the customer resulting in little choice regarding variance. All product and manufacturing specifications are in principle fixed to standard documents, which are retrieved whenever needed. Elements of this kind of processes can be found in combination with the previously mentioned processes. Select-to-order systems are in principle databases combined with salespeople and engineers that skillfully select the right product variant and related specifications during the order handling. For many manufacturers this is the most commonly used kind of specification process. Most consumer goods still belong to this class, for instance furniture, kitchen equipment, cars and bicycles [61].
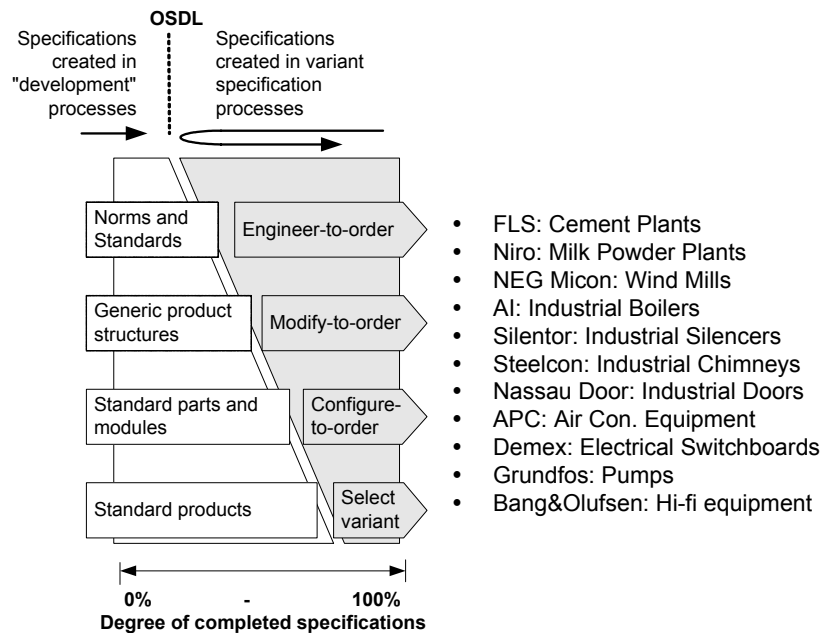
It must be emphasized that the four described types described above are found in combinations. The purpose has been to explain that variant specification processes can be very different, and that the OSDL is a fundamental characteristic, which can be used to describe this difference.

5. SETTING OSDL FOR FOCUSED VARIANT SPECIFICATION PROCESSES, -SOME PRACTICAL EXAMPLES

Classifying variant specification processes into engineer-to-order, modify-to-order, configure-to-order and select-to-order makes it possible to create an operational focus and achieve focused processes and systems, which better fulfill the needs of customers in order handling. Seen from a strategic point of view this concept of focused processes and system is similar to the classic thoughts of "the focused factory" [62]. By predefining different levels of the OSDL related to a specific product family it becomes possible to define preferred types of business processes, which target different customer needs and different performance measures. A company may choose to define e.g. configure-to-order as the preferred business process for a specific product family, and fit the core of operative systems, operative processes and information bases to support this. In this case more complex customer requirements should then be allocated to another process and system type, for instance engineer-to-order, which then again is focused and optimized to handle complexity and special customer treatment.

In the following there are given some practical examples of how to achieve focused processes through the development efforts. Each example can be related to a predefined level of the OSDL.

To achieve CTO processes a company has to develop predefined product parts and modules, with rules and calculation formulas for how to combine these. If a company desires to build electronic sales configurators, product configurators or production configurators a necessary preparation is thus to explicitly define and structure product and product related information in a way that supports a relatively simple configure-to-order task. This will not be possible if the OSDL is still at the position typically found in engineer-to-order.

In several cases it has been observed [61] that companies want to build configurators to support engineer-to-order or modify-to-order processes. This is a complicated task that often fails. Most companies in this situation experience that they must at first seek to move the OSDL towards the customer, thus reducing the complexity of the variant specification task. One way to illustrate this is by regarding several engineering companies who have built configurators for quotation processes. In order to be successful some of these (e.g. FLS-Industries, Aalborg Industries and GEA-Niro) have to some extent defined "phantom" modules and parts to be used for quotation processes, i.e. they do not match the actual engineered solutions 100%, but each "phantom" part covers many possible variants within 95-99% of the design options. These predefined parts and modules make it possible to configure a plant from building blocks, which do not actually exist, but which are so close to the real thing that for quotation purposes this is sufficiently. In order to make this possible the company will first have to spend some resources on the development of more standardized specifications prior to order acquisition, i.e. moving the OSDL considerably towards the customer.

One tool which is found more or less explicitly in manufacturing companies with CTO processes is assemble-to-order models (ATO) [51]. These models do in fact move the OSDL towards the customer, by explicitly defining a gross-list/super-list of subassemblies from which a final product may be configured. Such ATO models typically consist of a top-level model type (a product family) consisting of a list of classes, each containing all the possible subassemblies to select from. At the lowest level, which is not changed in the CTO process is the parts specified for each subassembly. In addition to the items in the ATO model one will have to make explicit the rules and calculations for how to select the right subassemblies. Strictly speaking it is impossible to have CTO processes without such ATO models for the product family structures.

ATO Model	Option class	Option Class Item	Parts
Model	Class 1	Item 1	
		Item 2	
		Item n	Part 1
			Part 2
			Part 3
	Class n	Item 1	
		Item 2	
		Item n	Part 1
			Part 2
			Part 3

Figure 4: Structure of an "assemble-to-order model".

To achieve efficient modify-to-order processes one will have to define structures similar to ATO models. However in this type of products it is not possible to define all parts and subassemblies in fixed part numbers before the order intake. Since the geometry is allowed to change new parts and modules will be specified during variant specification processes. If everything is allowed with regards to geometries one may characterize the task as engineering. However, the task can be simplified by pre-defining generic geometry structures with parametric dimensions, which are related to one another.

For instance it is possible to define the generic structure of an industrial silencer (figure 5) with a generic outer geometry and generic features, such as inlets and outlets. By creating parametric dimensions which all relate to relatively few top level parameters will make it possible to create new geometries and thus new parts relatively easy. Through this kind of parametric modeling it becomes possible to reduce the complexity of the engineer-to-order processes. The OSDL is in principle moved towards modify-to-order through a development effort made prior to order intake.
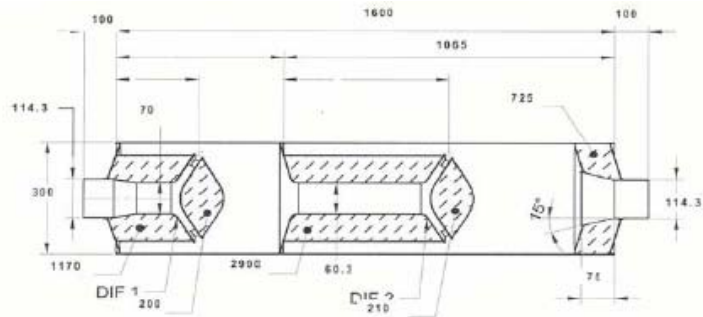

Figure 5: Main dimensions of an industrial silencer. This kind of products are typical specified through "modify-to-order" processes.

6. CONCLUSION

The present paper briefly presents an important issue related to variant specification processes in BTO companies, namely that of separating innovative long term development activities from order related day variant specification processes.

The proposition laid open for discussions in this paper, is that it may be useful to extend the traditional concept of CODP to an information/specification focused concept named OSDL. This concept can then be used in discussions of how to focus variant specification processes and how to separate information (specifications) developed prior to order intake from information made during order acquisition and order fulfillment. The concept of OSDL can thus be used to decide upon different levels of variant specification processes such as engineer-to-order, modify-to-order, configure-to-order and select-to-order. Some examples of how to achieve this focus is presented, for instance the use of ATO models and parametric design. Similarly OSDL levels can be used to discuss the necessary preparations needed prior to building configurators.

In conclusion the paper proposes and describes some fundamental issues related to the task of variant specification in BTO production.

7. REFERENCES

51. Hansen, B.L., (2004). Development of Industrial Variant Specification Systems. Ph.D. dissertation from the Technical University of Denmark, Lyngby, Denmark.

52. Hvam, L., Have, U. (1998). Re-engineering the specification process, Business Process Management Journal, Vol. 4 No. 1, 1998, pp. 25-43.

53. Anderson (2003). Build To Order And Mass Customization.

54. Pine, J. (1993).Mass customization, The New Frontier in Business Competition, Harvard Business School Press, Boston, 1993.

55. Hammer M., (1990), Reengineering work: Don't automate, obliterate, Harvard Business Review, July-August, 1990.

56. Barfod, A.,Hvolby, H.H. (1997). A. (InDanish) Ordrestyring, tidens indsatsområde (Order management, the focus area of our time), Industriens forlag, Denmark.

57. Pagh, J.D., (2000). (In Danish) Udvikling af virksomhedens transformationssystem (Development of the transformations system of the company), Ph.D. thesis from DTU, 2000.

58. Dam, A., Riis, J.O. (1986). (In Danish) Udvikling af styringskoncept (Development of a control concept), Institut for Produktion, Aalborg Universitetscenter, Driftsteknisk Institut, Institut for Produktudvikling, Danmarks Tekniske Hoejskole.

59. Shafer, N., Meredith, C., (1998). Operations Management, John Wiley and Sons, Inc. New York.

60. Sartori, L.G. (1988), Manufacturing Information Systems, Addison-Wesley Publishers, Great Britain.

61. Hansen, B.L., Riis, J.R., Hvam, L., (2003). Specification process reengineering: concepts and experiences from Danish industry. 10th ISPE International Conference on Concurrent Engineering, Madeira, Portugal.

62. Skinner, W. (1985), Manufacturing, the Formidable Competitive Weapon, Harvard University Graduate School of Business Administration, John Wiley & Son Inc.; New York.

PRODUCT CONFIGURATION SYSTEMS
-
IMPLICATIONS FOR PRODUCT INNOVATION AND DEVELOPMENT

Kasper Edwards and Jørgen Lindgaard Pedersen

Department of Manufacturing Engineering
Technical University of Denmark
Building 423, Produktionstorvet
2800 Lyngby
Denmark
Email: {Edwards@ipl.dtu.dk, jlp@ipl.dtu.dk}

Product Configuration Systems (PCS) is a step in the direction of mass customization in the sense that PCS allows a firm to significantly lower the unit cost of configuration. Thus PCS is a valuable technology for lowering operating costs while retaining a high number of possible product configurations. However, costs are but one parameter on which firms compete and firms must continually innovate new and develop existing products.
This paper presents original empirical insights on implementation and use of product configuration systems in a number of Danish industrial firms. The paper discusses the organisational changes associated with PCS and how this affects product innovation and development.
The paper begins by introducing product configuration systems, which are then placed in context to the firm as a process technology which coordinate different processes: product development, order acquisition, order fulfilment and production.

1. INTRODUCTION

Mass customisation has indeed become an important issue for many firms. Customers have become accustomed to the price of mass produced goods and are beginning to demand that products are customised to their needs while retaining the price associated with mass production. In Denmark firma are also pressured to deliver mass customization although Danish firms are characterised by an overweight of small to mid-sized companies, which traditionally have excelled in small batches and one of a kind production. However, like the rest of Europe Danish firms must follow suit, cut costs and deliver the desired products in which case mass customisation becomes a critical issue.

A means for firms to achieve mass customization are product configuration systems. A product configuration system consist of a computer model of a product, which contain information about the relationship between the individual components of the product and noteworthy any restrictions which one component impose on another. For instance a product model of a bicycle would have information regarding the frame, wheel, tube, tires, saddle, color, style of the different components etc. Restrictions in the model define what size of wheel fits with a give frame – no use in mounting a 26" wheel on a 12" frame.

The idea of mass customisation is little over 25 years old, beginning with Davis (1987) it has been research extensively since, see Silveira et al. (2000) for a literature review.

However it is only in recent years that off-the shelf software become available and many ERP systems now integrate configuration as a separate module in the system. So, things are dandy and firms should go ahead and achieve mass customisation with the available technology. Well, there are significant costs associated with achieving the potential benefits of a fully integrated product configuration system and this is exactly the theme of this paper.

The objective of this paper is to report preliminary results from a study of Danish Firms having implemented or are currently in the process of implementing product configuration systems. The project was initiated February 1st 2003 and will be terminated ultimo December 2004.

In the following paragraphs we will, more detailed, explain what a product configuration system is and place in a mass customisation context, followed by a section briefly describing the project, study and methodology. This is followed by a description of the involved firms and the results which are subsequently discussed and concluded.

2. PRODUCT CONFIGURATION SYSTEMS

A product configuration system is a computerized model of a product, which allow a sales person or customer to interact with the model and thereby choose the (configure) the desired components for this particular product. A product configuration system contain knowledge about the individual parts of the product and what variants of a part that are allowed to go together. For instance a product configuration system (PCS) for a bicycle will contain knowledge about the different sizes of frames, wheels, and various colours and their relationship. When a sales person configures a bicycle the PCS automatically ensures that the right size of wheels is added to the frame along with other items directed by the particular frame.

Product configuration systems can be designed with various levels of detail depending on the intended purpose. It is meaningful to adopt a distinction between 1) a tender configuration system and 2) a production configuration system. A tender configuration system is a PCS designed to produce a tender for a particular type of projects. Most often tender configuration systems are found in heavy engineering companies where the cost of producing a tender is very costly and a significant proportion of the tender contain the development of a product variant not previously developed. In the opposite end of the scale we find the production configuration system, which has complete knowledge regarding the product and its components down to the last spacer. Such systems are often linked to the company ERP system allowing a configured product to be directly produced. The PCS then feed a configuration to the ERP systems thus allowing for automation in the process of creating bill of material, routing, and inventory etc

A firm can be characterised as consisting of three distinct processes: 1) Product development, 2) Order handling and 3) Manufacturing as illustrated in Figure 26. We limit out focus to the development and order handling process and the relation in between.
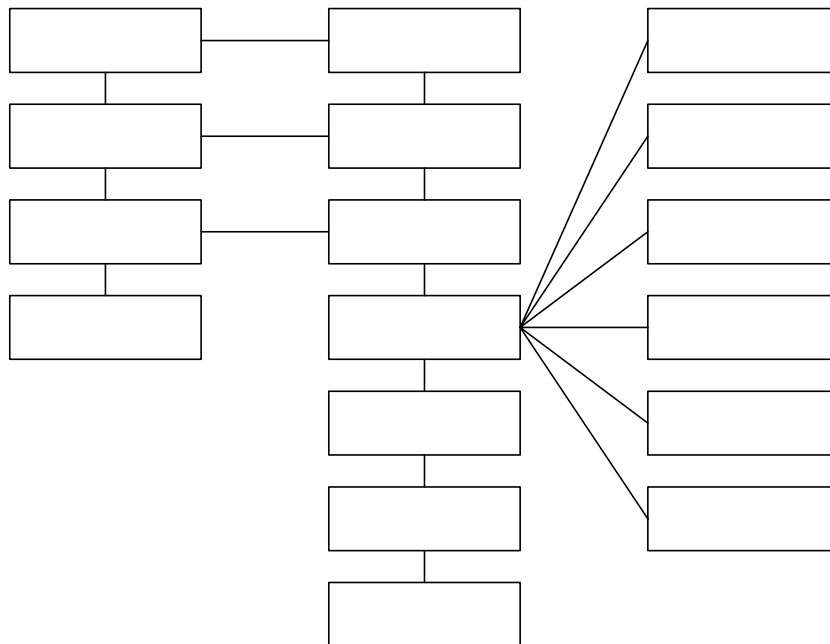

Figure 26: The process of a firm (Hansen 2003, from Barfod & Hvolby 1997).

Without PCS a customer enters in the top of the order handling process by contacting the firm. Sales personal and the customer enter into a dialogue with the intent of producing a product specification. If the product specification is outside normal parameters the product has to be modified to fit the customer's wishes. This is a costly procedure as the sales person has to contact product development who will make the proper adjustments. Regardless of how small these modifications

are, they have to follow the process of making product modifications. This includes making new drawings, bill of material and specifying the modified production process. All of these activities include approximations by the involved staff as they try, to the best of their abilities, to match the customers need to a specific product. Often the process is a reoccurring iteration between customer, sales, product development, and production. This is complex process where responsibility for the process is changed back and forth between the involved departments. All too often required information to specify the product was not gathered in the first place resulting time consuming follow-up questions, which could and should have been avoided. The questions do consume a whole lot of time but they halt the process of producing a specification and so do all the iterations between development, production etc.

The purpose of a PCS is to automate a substantial amount of these iterations by creating a system that allows a sales person to specify a product in dialogue with the customer. If the desired product can be specified by the product configuration system all the usual back and forth between departments is eliminated and the specified product can be produced when the customer has accepted the offer made by the firm.

A product configuration system extends the range of standard products and makes it possible to configure these at a low marginal cost. Without PCS limited experience of sales people result in only a limited variation of the possible configurations being sold. With PCS it is further possible to support the actual sales process by allowing the PCS to make choices regarding configuration based on functional requirements. For instance, if one was to configure a truck and the customer required a specific capacity the PCS would allow only configuration which satisfy the capacity requirement. Naturally a product configuration system does not support arbitrary configurations and the firm must choose which products and components to enter into the model. The possible number of configurations offered by the model becomes a subset of the configurations offered by the company, as illustrated in Figure 27.
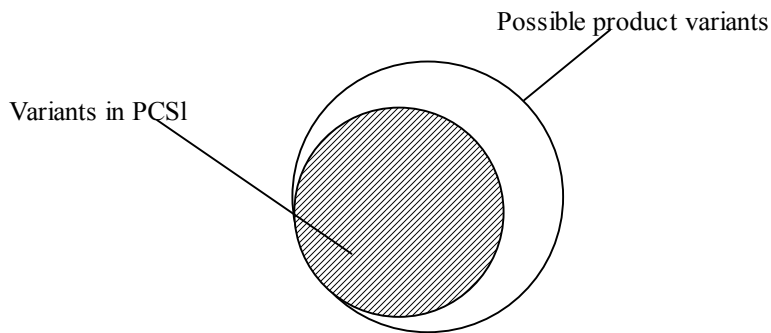

Figure 27: Possible product variants in a product configuration system is a subset of all product variants offered by the company.

While this may sound like the solution to many of our productivity problems it must be observed that there are significant problems with PCS or perhaps more specifically the implementation of PCS. It is evident from several implementation projects (Riis 2003, Hansen 2003, Hvam 1999) that there are significant costs associated with the implementation, and realizing benefits is dependant on several factors other than mere technical issues. Implementing PCS requires above all very specific insight into the product configuration process. It is imperative to know how and why a sales person or production employee configures a product as they do. What are the criteria for choosing one component in favor of another and this knowledge is most often tacit. But in order to develop a PCS this knowledge has to be made explicit and products must be grouped into different categories and the possible variation of elements specified.

For some firms this knowledge is tacit to the level where one is astonished that products are actually being produced. This observation is typically associated with firms producing one-of-a-kind products or small batches and is not related to firm size. Firms engaged in mass production typically have very explicit knowledge regarding their products as reducing the cost of a single part is important when competing on price.

Theoretically product configuration systems are part of the mass customization debate. Here we define mass customization as consisting of by two dimensions (Duray et al. 2000, p.607): 1) The basic nature of customisation, and 2) The means for achieving customization at or near mass production costs. The basic nature of customization refers to the observation (op. cit.) that variety in itself does not constitute customization At some point the configuration process the customer must provide input to the product specification. The means for achieving mass customization at or near mass production costs are essentially economics of scale as a consequence of modularity of the product.

When dividing a product into a number of distinct modules and making sure that some of these modules span more than just a single product it becomes possible to achieve economics of scale. Product configuration systems fit in the second category as a means for achieving customization at little marginal cost. However, it does not follow that the cost of producing the different variants will be the same. As Pine (1999, p196) notes: "The best method for achieving mass customization – minimising costs while maximising individual customization - is to create modular components that can

be configured into a wide variety of end products and services", which is also recognised by Duray et al. (2000, p608). While it is easy to design a product configuration system around a fully modular product, it is not a necessity, and it is possible to design a product configuration system for a non-modular product. The latter product will not see the cost advantages of modularisation, and the process of creating the configuration system will also be more complex due to idiosyncrasies in the individual product variants.

3. METHODOLOGY

The research reported in this paper is but one result from the "The Product Models, Economy, Technology, and Organisation" project (PETO). The PETO project is studying the process and effects of implementing product configuration systems, which so far has received little attention by scholars. Most of the literature on PCS is centered in the mass customization debate and focuses on the technical aspects of configuration. However, research is emerging, which take into account more than just the technical issues and in particular Forza & Salvador (2001 & 2002) has offered insight into the economic consequences of PCS. Still the organizational perspective is relegated to an anecdotal level and general and long term effects are not discussed.

As no other interdisciplinary studies on PCS have been conducted, an explorative and hypothetical deductive approach was selected. Based on knowledge on product configuration systems a number of hypotheses was deduced for each of the economic, technical, and organization perspectives. This was then transformed into a questionnaire populated with questions directed at: 1) The specification process before and after implementing product configuration systems, this is the foundation for understanding the changes induced by implementing a product configuration system; 2) Technical issues of the implemented product configuration system; 3) Economic issues and 4) Organizational issues. The questionnaire was designed to document the order acquisition chain before and after implementing PCS. The questionnaire consisted of 196 questions of which 47 were directed at economic issues, 33 at technical issues, 97 at organizational issues, and 19 regarded the specification process. The questions were designed to be both closed and open ended questions, in the latter case leading respondents to elaborate and explain certain positions (Jacobsen 1996:111). The open ended questions were used deliberately to allow some degree of exploration in the interview process, and respondents were allowed to pursue their line of thought before being interrupted and directed towards the question. Concluding questions were used to confirm and summarize the meaning of open ended questions.

17 firms among the members of the Danish Association for Product Modelling[23] were approached and 13 firms agreed to participate in the study, however, one of the 13 firms did not have PCS implemented and was interviewed only because they had rejected to implement PCS. Thus 12 firms with product configuration system experience were subject to qualitative interviews using the developed questionnaire.

Firms were asked to provide individuals of the following categories: 1) Sponsor, 2) Technician/programmer, 3) User, and 4) The project leader. These four roles were chosen, as they would theoretically represent all organizational levels of a product configuration project.

The interviews were intended to be conducted with a single respondent at a time, allowing for a detailed interview with personal opinion expressed. To help establish a minimum level of trust, respondents were provided with a written and signed statement expressing that the information would remain anonymous and certainly not shared with their colleagues.

However, in some cases it was not possible to conduct individual interviews, and a group interview was the only option for having the specific firm participate in the study. It is expected that these interviews to some degree fail in uncovering problems with the product configuration system and the implementation process, which are due to personal differences. Group interviews have a tendency of expressing consensus among the respondents.

In all interviews multiple investigators (Eisenhardt 1989:538) were used to ensure complementary opinions and insights and to enhance confidence in the findings. During all interviews two investigators were present, and on some occasions even three and four investigators found their time to participate in the interviews. The combination of multiple investigators and open ended questions is very powerful, if investigators deliberately keep silent to pressure respondents into answering. On many occasions this was the deciding factor for getting a meaningful answer. The interviews were taped and subsequently transcribed, which was followed by a condensing procedure for extracting the meaning of the interviews (Kvale 1994:189).

4. EMPIRICAL EVIDENCE

12 firms have been interviewed all of which had or was in the process of implementing PCS. The participating firms have all been promised to remain anonymous and we shall refer to them as firms A through L. The cases will be divided in two

---

[23] See www.productmocels.org.

groups: Those offering evidence supporting our claim and those not. Two of each type of cases will be presented in detail and the remaining eight cases will be briefly summarised.

Firms A and B manufacture production plants and single orders are often seen in the 100 Million Euros range. Firms A and B are large and old firms with a lion's share of their market. Both firms are engaged in heavy engineering and the bulk of their resources are spend on engineering hours. Firms C to G consist of firms which, which all has a history of being mass producers. Firms C and D produce consumer goods and firms E, F and G are industry suppliers, the turnaround in this group range from 12,5 to 600 Million Euros. Firms H to L are firms which are batch producing firms, which allow the firms to modify the product specifications. The turnover in this category was between 22 and 550 Million Euros.

Cases Supporting our Claim

Firms supporting our claim are: A, B, E, F, I, and J.

Firm A

Firm A is a heavy engineering firm which implemented PCS for two reasons: 1) Maintain key knowledge within the company and 2) Make the product specification process more efficient. Firm A is in a situation where a significant number of the key staff is due for retirement within the next 2-5 years and much of their knowledge is not documented. As this staff is highly specialized engineers this is a potential threat for the company. Heavy engineering firms is characterized by the fact that a specific tender for must be produced for each single client. Firm A has further observed that the number of tenders produced for each order is rising and is currently 33 tenders' per order. Firm A is currently in the final staged before putting their PCS into productive use. However, the experience is that firm A has developed a PCS which meet the original criteria and thus Firm meet the first part of our claim.

It is, however, apparent that the product configuration system will have unforeseen effects on the innovation system in firm A. Before PCS most product development was a direct result of customer requests and when similar products has been developed a number of times they were regarded as standard products. PCS is now envisioned to be an integral part of product development where new models are developed within the product configuration system before customers express a need for the product. This also changes the role of development from reactive to proactive and new ideas for products must be generated using other sources of inspiration than customer input. Firm A support the second part of our claim by demonstrating that PCS has induced a change in the perception about how products are developed. PCS is not believed to be a central part of future product development and this was not envisioned in the beginning of the project.

Firm B

Firm B, implemented PCS with the purpose of reducing the amount of resources required for the production of a tender. In firm B a tender requires an average of 2500 engineering hours and such workload often drain the firms resources in times where many customers approach. Firm B has traditionally dealt with this situation by estimating the probability of getting the deal and rejected others. However, it appears that the procedure for predicting the customers to address is not functioning as a review of past orders and estimates revealed less than 50% hit rate - in other words they felt it impossible to make a prediction. This called for a tool, which could automate the process of generating a tender and allow the firms to produce tenders for all interested potential clients.

Firm B went through a development process where the actual work processes and products were analysed and documented (see Riis 2003, p.186 for a general description of such an analysis). The product configuration systems were implemented and firm B experienced that the PCS was able to deliver the expected benefits. Firm B experienced that a tender of higher quality i.e. more precise in responding to specific customer wishes, could be produced in an average of 190 engineering hours compared to the previous average of 2500 hours.

Thus, firm B support the first portion of our claim: A tool has been selected and implemented for a specific purpose of automating the tender producing process. Firms B also went through a significant process of restructuring the organisation and their perception of their products, which was not anticipated.

Firm B is perceived as an engineering firm and proud of it. Consequently it is has been corporate culture to cater for the individual customers by producing a product which was tailor made. If the customer needed a grinder 7.5 meter in length the customer would get it regardless of the 8.0 meter grinder developed just last week. When a number of customers had requested similar products the product would slip into the catalogue as standard product. Such development has significant implications for the costs of the product and when department A changed their grinder department B and C etc. would have to adjust accordingly and this is very costly but consequences was invisible to department A.

In the process of developing PCS it became necessary to break the product in to categories based on production capacity of the product. Consequently the various departments were asked to deliver estimated costs for a variety of their components, such as the grinder. It became chokingly clear that some variation in cost was not tied to whether the grinder was 7.5 or 8.0 meters and actually a majority of the need for grinders could be satisfied in just three different models. This was done for

all of the components in the product and it was possible to configure the PCS to produce a tender based on a information proceed by the customer in the first interview. The tender produced could then serve as a basis for further negotiation.

Once delivered the individual departments were forced to deliver components which complied with the specifications and this resulted in a significant modularisation, which is changing the perception of how to develop the product. Departments now think in modules, which are structured according to an overall design of the factory e.g. capacity of the plant. This change is a fairly recent event and has happened during 2002 and 2003. For this reason it is not possible to clearly identify the effects on innovation from modularisation. It does however remain clear that the nature of innovation has changed. Before PCS innovation was part of every order but now PCS ensure that the same modules are used many times. It is not clear what the source of innovation will be now and no alternate means of stimulating innovation has yet been determined. Therefore we can fear that innovation is slowed down with possible negative long term consequences. The conclusion is that innovation is affected but it is not yet possible to determine how and what the consequences are.

Firm E

Firm E needed a tool to reduce the turnaround-time for specifying a new product variant and succeeded in achieving a reduction from 6 to 1 day.

Firm E was force to change their product development practice from reactive to proactive. Before implementing PCS customers would demand changes made to an existing product e.g. improved resistance to high temperature conditions. The development department would then make the proper modifications and after seeing the same demand the product would become a standard product. Now, the development department act proactively and predefine a number of variations when developing a new product.

Firm F

Firm F experienced that the quality of product specifications was too low and needed a tool to validate product configurations. PCS was implemented as a tool to improve the quality of specifications and it was a great success as the number of faulty specification has dropped from 25% to 0%, measured as the number of production halts due to faulty specifications - quite remarkable. This supports the first part of our claim.

Before implementing PCS most product configurations had to pass through the product development department for validation. After implementing PCS most of the validation is done using the PCS and now product development can actually focus on developing new products. In the situation a product configuration is not validated through the PCS it must be validated in the product development department. There are both advantages and disadvantages to this change. The obvious advantage is that the majority of product validation is now done automatically which has resulted in lower turnaround time. The disadvantage is that the product development department is rarely presented with odd customer requests generating fewer ideas for new products. There has also been a change in focus in product development. Product development now focuses primarily on cutting costs and new models are often a redesigned version of a previous model e.g. the model is a direct replacement in terms of size and quality but significantly cheaper due to changes in assembly and production routines. Another change is the facts that firm F now only embarks on developing a radical new design or product if such a product is backed by a customer who is willing to be part of the development and buy a certain amount.

These are all changes to the innovation process which has been facilitated by the product configuration system and we conclude that firm F supports both the first and second part of our claim.

Firm I

Firm I implemented PCS as an integrated part of a new ERP suit and needed PCS specifically to establish a clear overview of the product range. Firm I had a history of producing tailor made products for individual clients. Firm I produce a product, which is a productive resource for the customer and the yield of the product is directly related to its height. Needless to say customers demand the highest product available. However, the same relation does not exist between production cost and height, which has lead to a number of unpleasant surprises in terms of lost profit. The combination of PCS and ERP has resulted in a more managed production process where only valid products can be configured and sold.

The use of PCS has lead to insight into the structure of all the different models produced by firm I. It has been discovered that there are significant potential for modularization e.g. a cooling system of equal effect is not the same for different models. The result has been different housing and cooling hose systems, which further makes it impossible to share parts between models. The overview provided by the PCS has resulted in a move towards a more modular type of development whereas earlier the process has been typically integrated product development, which supports the second part of our claim.

Firm J

Firm J produces equipment to be used on trucks and the product has to be configured to particular type of truck used by the customer. The functional characteristics of the product also have to be specified and naturally there are many restrictions

between the functional and structural specifications. Before implementing PCS only a single individual was able to thoroughly validate product specifications and calculate a price.. In most instances sales staff would pass a product specification for validation and it would contain no errors but occasionally errors were detected. It also happened that products were validated and later in the production stage it was realized that product simply could not be produced. An increase in the number of orders resulted in the person doing the validation and price calculation becoming overloaded leading to prolonged response time. Firm J choose to implement PCS as a means of overcoming this particular situation and PCS was to ease configuration and allow sales staff to determine the actual price. This supports the first part of our claim.

The PCS, however, also had a significant effect on the process of innovation. Before PCS products were developed in the same pace as customers desired variants not part of the standard products. The innovation process was typical integrated product development in the sense that functional innovation was specifically tied to the structure of a product. Thus, when a new customer demanded the same functionality but to be used on different equipment the product had to be developed all over again.

PCS had the effect that the existing range of products was systematically divided into a number of product families. In the process of doing so it was discovered that on many occasions the same functionality was provided in different ways i.e. the bill of material was not at all similar. At first this was just noted but not acted upon. After making the PCS available for both sales and product development a common understanding of the different product families evolved. The PCS provided a clear overview of the different products and their parts and this over time resulted in product developers beginning to apply a modular approach to new functionality. A modular approach has the distinct advantage that a specific functionality is no longer are tied to a specific product structure. This advantage becomes further pronounced when using PCS as the PCS immediately allows a feature to be used across structures as long as a particular structure has the required interface.

The change in product development strategy was not anticipated but indeed welcomed due to the inherent cost savings from not having to develop for a specific structure. Thus, this case also supports the second part of our claim.

Cases Not Supporting our Claim

Firms C, D, G, H, K and L do not support our two claims.

Firms C and D

Firms C and D are both manufacturing products in which the element of design is very important. Sales are driven by product design rather than functionality and specific features. These firms have deliberately decided design to be the governing factor in product development. This becomes evident in the organisation where the PCS department is involved only after the product has been developed and then only to build a model, which allows for easy configuration. These two firms also had long term experience with PCS in which case it cannot be ruled out that product development has been affected in the past although the respondents denied this. This leads to a conclusion, although based on only two firms, that when design is a dominating parameter it is possible to use PCS as a specific tool which can be implemented and managed without unforeseen consequences for innovation.

Firm G

Firm G implemented PCS as a tool with the single purpose of improving intra company sales and allocation of production resources. Firm G long time ago implemented PCS within their ERP system allowing sales staff to easily configure and allocate production resources to the order at hand. When firm G was interviewed they were in the process of implementing a new PCS as a replacement. In the new PCS all production resource across subsidiary companies has become available to all allowing easy allocation of resources. As the products already were configurable no new insight has been gained in the process. It is possible that the original PCS implementation has had significant effects on innovation although it cannot be substantiated.

Firm H

Firm H produces uninterruptible power supply systems and introduced PCS as a general concept for selling their products. Product configuration and allowing customers to be part of the configuration process is considered the core of their strategy. Product innovation is undertaken in the product development department and the configuration department program their PCS the moment a new product or variant has been released. Modularisation is also a core concept in firm H although it has been motivated from production and development rather than from configuration. Firm H emphasized modularisation and take great care to develop products that can be produced and assembled as cheap as possible. One example is a product family which has to major variants, which has 110 components in common. Variant 1 requires 25 additional components and variant 2 requires 46 additional components. While not motivated by configuration such product strategy is a perfect match with configuration as the required division into product families are already done in the development stage. Thus, firm H does neither support the first, nor the secondary claim.

Firm K

Firm K implemented PCS with the sole purpose of being able to make sense of an enormous amount of product variants. Firm K differentiates their product by offering custom sizes and not just a number of standard sizes. This results in an enormous number of variants and since the product produced by firm K has a very long lifespan, which is currently around 40 years for one product line, the possible number of variants is very large. PCS has not influence the number of variants, nor the product innovation process. Product innovation is driven by long term changes in the market and not possible modularisation. It would appear that firm K like firm C and D is driven by design but also give priority to reducing costs. PCS reduce costs by streamlining the specification process and producing ready to produce specifications.

Firm L

Firm L manufactures electric distribution boards for industry use and implemented PCS in order to allow customers to do their own configuring using a web browser. The general idea was to increase productivity by letting the customers do some of the work, so in this sense there was a clear objective for the project. Firm L succeeded in developing a PCS fulfilling the goals and it has not been possible to detect changes in production or innovation as a consequence hereof. In fact, it was part of the purpose that the sales department was to use the configuration system, which never happened. The sales department feels that the configuration system provides too many constraints and therefore they continue to use the old ERP based system.

5. CONCLUSION

This paper stated the claim that new technology is introduced in a firm as a simple tool based on a clear understanding of the positive effect on the firm. It is further our claim that the introduced technology, as an unforeseen side effect influence the innovative capability of the. The claim was tested against experience from a project analysing 12 Danish firms having implemented product configuration systems.

Six of the analysed firms support the claim and all exemplify that technology is chosen as a solution to a specific problem and the consequences are not completely understood. In all of the supporting cases PCS changed the order acquisition process making it more streamlined. The benefit was lower turnaround time and consequently higher productivity. PCS inspire modular thinking in the sense that in order to use PCS a model of the products had to be programmed into the system. Some firms went through a formal process for developing such a model and others just programmed their products one by one into the system. Regardless, all firms have experienced an increasing focus on modular development as compared to integrated development. Modular development has the advantage that discrete modules can be replaced without having to redesign the whole system. Needless to say this has positive effects on productivity. However, it is not possible to predict if these changes will have a long term positive or negative effect on the firms.

Of the analysed firms whom did not support our claims it is interesting to observe that three of the firms (C, D and K) produce products, for which design is the perhaps the most important issue when developing new products. These firms have implemented PCS to support the sales process, which in turn has had spill-over effects into production planning but not product development. Firms L, G and H all had different reasons for implementing PCS. Firms L and G already had experience with similar technology and both product and production processed appeared to match PCS.

Based on the evidence presented it appears that complex technology indeed do influence innovation and the six supporting cases where innovation has been influenced this was not anticipated.

6. REFERENCES

1. Davis, Stanley, 1987, "Future Perfect", Addison-Wesley.
2. Duray, Rebecca, Peter T. Ward, Glen W. Milligan, William L. Berry, "Approaches to mass customization: configurations and empirical validation", Journal of Operations Management 18 (2000) 605-625.
3. Empirical evidence from the PETO project (Product Models – Economics, Technology and Organisation). More than 30 interviews are planned and in the process of being conducted in Danish companies. Currently 9 interviews have been completed.
4. Edwards, K, Lars Hvam, Morten Møldrup, Niels Møller, Jørgen Lindgaard Petersen, Jesper Riis. 2003 "Teoretisk Foranalyse", The PETO project, Department of Manufacturing Engineering and Management, Technical University of Denmark.
5. Eisenhardt, Kathleen M., 1989, "Building Theories from Case Study Research", Academy of management Review, Vol 14, No.4, pp.532-550.
6. Faltings, B. and Rainer Weigel, 1994, "Constraint-based knowledge representation for configuration systems" Technical report No. TR-94/59, Department D'Informatique.

7. Forza, Cipriano and Fabrizio Salvadore, 2002, "Product configuration and inter-firm co-ordination: an innovative solution from a small manufacturing enterprise", Computers In Industry 49 pp. 37-46.
8. Forza, Cipriano and Fabrizio Salvadore, 2001, "Manageging for variety in the order acquition and fulfilment process: The contribution of product configuration systems", Int. J. of production economics 76, pp. 87-98.
9. Hansen, Benjamin Loer, 2003, "Development of Industrial Variant Specification Systems, Ph.D. thesis, Department of Manufacturing Engineering and Management, Technical University of Denmark.
10. Hvam, Lars, 1994, "Anvendelse af Produktmodelering – set ud fra en arbejdsforberedelsessynsvinel", Ph.D. Dissertation, Driftteknisk Inistitut, Technical University of Denmark.
11. Hvam, Lars, 1999, "A Procedure for building Product models", Robotics and Integrated Manufacturing.
12. Jacobsen, Jan K., 1997, "Interview – Kunsten at lytte og spørge", Hans Reitzels Forlag (in Danish).
13. Kvale, Steinar, 1997, "InterView", Hans Reitzels Forlag (in Danish) translated from English: Nake, Bjørn, 1994, "InterView"
14. Pine,B. Joseph II, 1999, "Mass customization – The New Frontier in Business Competition", paperback, Harward Business School Press.
15. Riis, J., 2003, (In Danish) "Fremgangsmåde for Opbygning, implementering og vedligeholdelse af produktmodeller", Ph.D. Thesis, Department of Manufacturing Engineering and Management, Technical University of Denmark.
16. Silveira, Giovani Da, Dennis Borenstein and Flávio S. Foglitto, 2000, "Mass customization: Litterature review and research directions", Int. J. of production economics.
17. Schwartze, S., 1996, "Configuration of Multiple-variant products", BWI, Zürich