A Multi-Agent based Configuration Process for Mass Customization

Blecker, Thorsten and Abdelkafi, Nizar and Kreutler, Gerold

2004

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 companyís operations, as well as a high external complexity level from a customerís 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 customizerís 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 customersí 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 customerís perspective and supporting managers in variety steering decisions which aim at reducing the internal complexity from a companyís 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 ÑÖ that nearly everyone finds exactly what they wantî (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 companyís and customerís 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.

Blecker, Th./Abdelkafi, N./Kreutler, G.: A Multi-Agent based Configuration Process for Mass Customization,
in: Department of Manufacturing Engineering and Management, Technical University of Denmark (Ed.):
Conference Proceedings, International Conference on Economic, Technical and Organisational aspects of
Product Configuration Systems, Technical University of Denmark, Lyngby/Denmark, June 28 - 29th, 2004, pp. 27 - 33

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 customerís preferences and profile in order to display only the subset of the relevant product variants. From the mass customizerís 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 supplierís success, which would suggest the retention of these variants in the mass customizerís 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 customizerís 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-‡-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 constraintsí 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 ìÖ software systems that guide customers according to their profile and requirements through a ìcustomizedî 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.î The advisory component ensures the online communication by initiating an advisory dialog, in which customers are asked targeted questions. Based on the customersí answers, the advisory component accordingly adapts the dialog in order to better elicit customer requirements. The advisory component maps a customerís 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 variantsí 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 customerís 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 customerís requirements to each selected platform agent.
- Communication of the module variantsí cost ranges to the auction agent. On the basis of these costs the auction functions are determined.

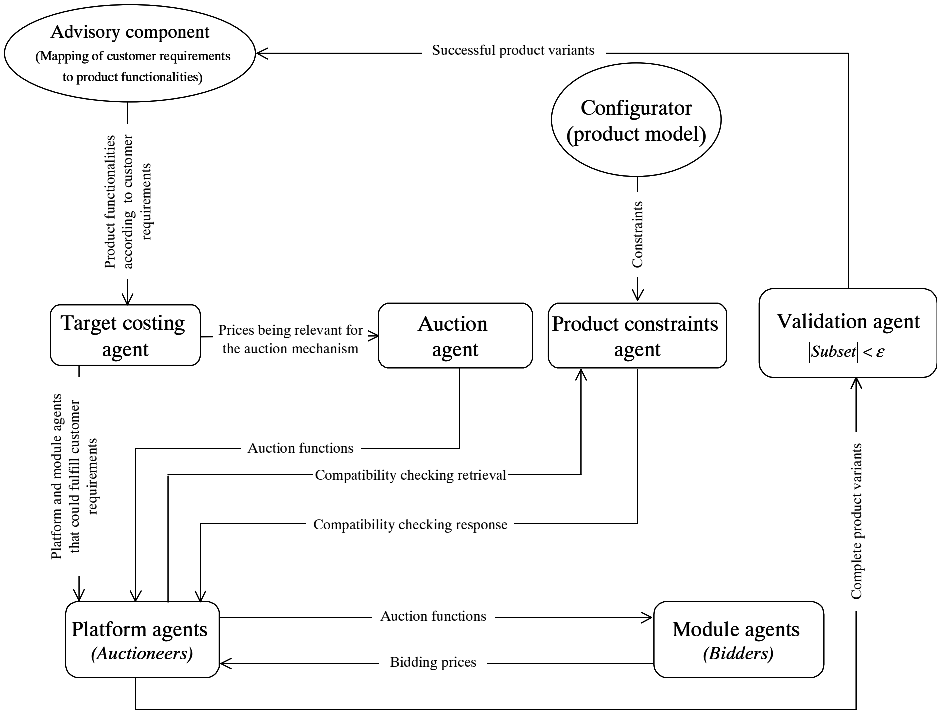

Figure 1: Framework supporting the multi-agent based configuration

In the outlined framework, the auction agent plays a relevant role. It receives the module variantsí 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 constraintsí 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 constraintsí agent must have direct access to the configuratorís 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 variantsí *subset* that consists of the end products with the best specifications to suit customersí 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 agentís 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 agentís account is supposed to linearly decrease in the course of time. When the module agentís account is equal to zero, it risks ìdeathî. 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 constraintsí 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 agentsí 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 AGENTSí BEHAVIOR FOR REAL-TIME CONFIGURATION

The bidding strategy of the module agent is formulated on the basis of the agentís 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 $\frac{Profits}{T_\infty}$ 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 $\frac{Profits}{T_\infty}$ 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 customersí 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.

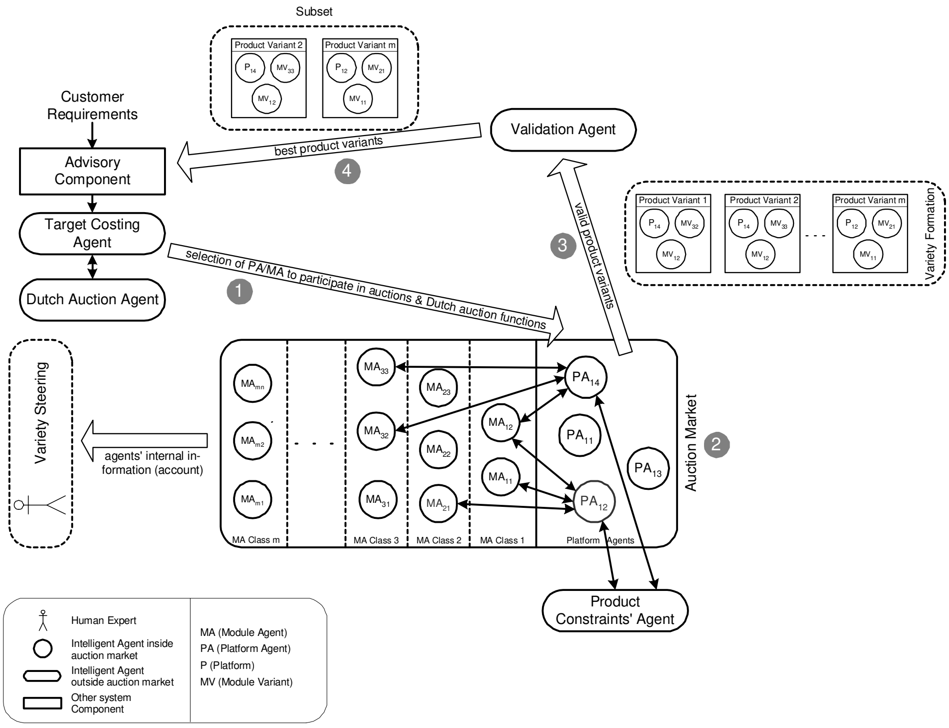

Figure 2: The multi-agent based configuration process

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 companyís 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 supplierís 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 customersí 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 customersí 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 constraintsí 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 Customersí 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.