DYNAMIC MULTI-AGENT BASED VARIETY FORMATION AND STEERING IN MASS CUSTOMIZATION

Thorsten Blecker, Nizar Abdelkafi
Departement of Production/Operations Management, University of Klagenfurt, Austria

Gerold Kreutler, Gerhard Friedrich
Computer Science and Manufacturing, University of Klagenfurt, Austria

Keywords: Product Configuration, Mass Customization, Variety Formation and Steering, Multi Agent System

Abstract: 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. To cope with both complexity problems, an information system based on agent technology is able to be identified as a suitable solution approach. 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 have to compete with each other in order to join coalitions representing salable 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. Furthermore, in order to integrate the multi-agent system in the existing information system landscape of the mass customizer, a technical architecture is proposed and a scenario depicting the main communication steps is specified.

1 INTRODUCTION

Mass customization is a business strategy that aims at satisfying individual customers' needs nearly with mass production efficiency (Pine, 1993).The development of mass customization is essentially 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 Internet based customer-supplier interactions can be considered as the most relevant enablers for a successful implementation of the strategy. Rautenstrauch et al. (2002) pointed out that information systems provide the necessary support for enterprises pursuing mass customization.

The information which arises during the interaction process between the customer and supplier serves to build up a long-lasting individual customer relationship (Piller, 2001). Due to high customer orientation, mass customization induces a variety-rich environment. However, customers generally do not seek out variety per se. They do only want the choice that fits to their needs.

The resulting variety in mass customization triggers a high complexity level that leads to additional costs. Moreover, because of the limited human information processing capacity and lack of technical product knowledge, excessive variety confuses customers who are overwhelmed by the complexity of the decision making process. Therefore, the main goal should be to find an optimal product variety which leads to the optimal cost-benefit-relation. For example, Blecker et al. (2003) propose a key metrics system to cope with the internal variety-induced complexity and emphasize the importance of the interaction systems to reduce the external complexity experienced by customers during the buying process.

From this point of view, we can identify two challenges. Firstly, the mass customizer must be supported by an information system to efficiently cope with variety. Secondly, it is relevant to assist customers with adequate information tools during the individualization process in order to lead them in a fast paced manner and with a low amount of effort to their optimal choice. In this paper, after a short description of the main variety problems in mass customization, we formally define a multi-agent based approach supporting dynamic variety-forma-

Blecker T., Abdelkafi N., Kreutler G. and Friedrich G. (2004).
DYNAMIC MULTI-AGENT BASED VARIETY FORMATION AND STEERING IN MASS CUSTOMIZATION.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 3-13
DOI: 10.5220/0002608400030013

tion and steering enabling mass customizers to face both depicted challenges. Then, we suggest a technical infrastructure for the implementation of the multi-agent system.

2 VARIETY PROBLEMS IN MASS CUSTOMIZATION

Due to the high individualization level in mass customization, final products are not manufactured until a customer order arrives. This customer-pull system improves the planning situation in dynamic markets and avoids costs such as those due to final products' inventory and special offers to incur. However, the huge variety induced in mass customization is associated with a high complexity and involves costs which arise in the form of overheads. Rosenberg (1997) mentions that product complexity is essentially due to two main reasons which are (a) the variety of product types and groups and (b) the number of components being built in the products, as well as their connections with each other.

An empirical study of Wildemann (2001) has shown that with the doubling of the number of product variants, the unit costs would increase about 20-35% for firms with traditional manufacturing systems. For segmented and flexible automated plants the unit costs would increase about 10-15%. Wildemann concluded that an increase of product variety is associated with an inverted learning curve. Furthermore, Rathnow (1993) depicts a huge product variety is not usually profitable and points out that there is a point $V_{opt.}$ (optimal variety) from which the cost effects of product variety overcompensate its beneficial effects. Lingnau (1994) qualitatively examines cost effects which are involved when increasing variety. He considers a functional organization structure and scrutinizes the effects of variety on sales, production, purchasing and research and development. Lingnau points out that variety generates additional costs in each function. For example, when introducing new variants, new distribution channels could be necessary. Increased variety also complicates the production planning and control and more setups leading to longer idle times in which are required. With higher variety the work-in-process inventory also increases and quality assurance measures should be intensified.

The introduction or elimination of product variants are decisions which are made within the scope of variety steering. Blecker et al. (2003) make the distinction between variety management and variety steering. Variety management embraces the concepts that can be applied in order to increase component and process commonality levels during a company's operations such as part families, building blocks, modular product architectures, etc. Unlike variety management concepts, variety steering concepts essentially deal with external variety, which can be perceived by customers. In this paper, we assume that the mass customizer has already implemented a variety management concept and that the main decisions concern variety steering.

The excess of variety and the resulting complexity can endanger the success of mass customized products whose prices should not dramatically differ from the corresponding ones manufactured with a mass production system. That is why, it is relevant to efficiently cope with the internal effects of variety in mass customization. In addition to high internal complexity level during a company's operations, variety induces external complexity that has bad effects from a customer's perspective.

Due to the limited human information processing capacity, excessive variety could confuse customers. Furthermore, customers are not aware of their needs until they see them violated. By looking for suitable products in huge assortments, customers can experience stress, frustration or regret. Iyengar/Lepper (2000) also claim that in limited-choice contexts people are engaged in rational optimization, whereas in extensive-choice contexts people simply end their choice-making when they find a choice that is merely satisfactory, rather than optimal. Furthermore, Schwartz (2000) indicates that by adding new options, the choice situation would be less rather than more attractive and that some people would look for the help of e.g. experts, who make the decision for them.

On the one hand, in order to avoid customers getting lost in huge product assortments, the mass customizer should support them during the interaction process to help them find the product variants corresponding to their optimal choice. On the other hand, the mass customizer has to strongly consider the internal complexity by efficiently steering variety. Therefore, a comprehensive solution approach must integrate both customer's and supplier's perspectives in one information system.

3 A MULTI-AGENT APPROACH FOR VARIETY FORMATION AND STEERING

Common configuration systems for mass customization necessitate specific product knowledge and often overstrain customers. Therefore, we are convinced that these systems should be improved to better support customers during the elicitation process. Blecker et al. (2004) opt for interaction systems which are capable of assisting customers through advisory. Thus, the interaction system should be able to capture a customers' preferences and profile in order to display only the subset of relevant product variants which would better fulfil customers' requirements. From the huge product assortment, only the best variants succeed to be short-listed and displayed to customers. Consequently, in the long run these will better contribute to a supplier's success. Those which are not short-listed will only trigger high complexity and are not relevant for customers. This would suggest that the product variants would compete with each other. That is why, it is necessary to define a mechanism setting the rules which organize the competition between variants. This leads one to consider a market mechanism supported by multi-agent technology. The complexity and fuzziness of the problem are further reasons for the use of a multi-agent approach.

The multi-agent based system should dynamically support each user during the interaction process. This means that the system should iteratively generate and refine product variants according to specific customers' needs. Concurrently, it supports the long term supplier's variety steering. This is realized by the decentralization of variety decisions which are supported by autonomous agents.

At first, we present the assumption and definitions required to build up the multi-agent system. Then, we conceptually describe how agents can reach agreements in order to form product variants.

3.1 Assumption and Definitions

Pine (1993) pointed out that the best method to achieve mass customization is to develop products around modular architectures. Ericsson and Erixon (1999) defined modules as building blocks with defined interfaces. By combining only a few modules, it is possible to construct a huge number of product variants. The economies of scale are reached through modules instead of products and economies of scope are attained when modules are built in different products. That is why the assumption of this paper is as follows:

Assumption: Modular product architecture

We assume that the complete set of product variants can be manufactured on the basis of modules.

The main idea is to consider the module variants to be built in the different product variants as autonomous rational agents. It is more reasonable to consider the module variants as agents than the product variants because with a few modules, one can build up a very large number of product variants which can go up to billions. Thus, by considering modules as agents the problem remains manageable and the computational resources are not overstrained. Therefore, we provide the following definition:

Definition 1: Module agent

Let M be the set of all modules, $M = \{M_1, M_2, \ldots, M_m\}$. We call M_i a module class. A module class M_i contains a set of module variants $MV_{i1}, MV_{i2}, \ldots, MV_{ip(i)}$. p is a function associating an index i of a module class with the index $p(i)$ referring to the number of module variants in a module class. With each module variant $MV_{ij}, j = 1, \ldots, p(i)$ we associate an autonomous rational agent, called a module agent $MA_{ij}, j = 1, \ldots, p(i)$ which disposes of resources and is able to perform tasks.

Modules can be classified in must- and can-modules. Must-modules are indispensable for ensuring the basic product functionalities, whereas can- modules are optional. For example, an engine is a must-module for a car. Without an engine a car cannot ensure its basic functionality consisting of mobility. In contrast to the engine, an air-conditioner is a can-module because it does not disturb the main functionalities a car must perform. In this context the term platform is defined in the technical literature in two distinctive views:

- A product platform is the set of all modules required for the manufacturing of all possible product variants (e.g. Ericsson/Erixon, 1999).
- A product platform can also be the appellation of a specific *common module* which is used in a great range of product variants (e.g. Piller/Waringer, 1999; Wildemann, 2003). This definition is commonly used in the automobile industry.

The second definition of platforms will be adopted in this paper because it considers the platform as a module having an additional relevance in comparison to other modules, which is mainly due to its implementation frequency in product variants. The corresponding agents are called platform agents to make the distinction vis-à-vis other module agents. Furthermore, the set of all platform and module agents are grouped in an agent pool. All different agents are members of a multi-agent system whose main goal is to present only a *subset* of product variants, which would best fit customers' needs.

Because only a subset is allowed to be displayed to customers, the product variants have to compete with each other. Due to the modular architecture of products, we can argue that the module variants also compete to be existent in the set of the presented product configurations. Being driven by a further motivation of this work to support variety steering decisions in mass customization, the module variants which do not resist competition should be eliminated. Therefore, it is legitimate to provide the second definition:

Definition 2: Self-preservation
Each module agent MA_{ij} strives for ensuring its existence by having enough resources to survive.

Definition 2 leads us to consider evolutionary theory which sees evolution as the result of selection by the environment acting on a population of organisms competing for resources. The winners of the competition, those who are most fit to gain the resources necessary for survival, will be selected, the others are eliminated (Kauffman, 1993). The resources of an agent are stored in an account which is defined as follows:

Definition 3: Module agent's account
Each module agent MA_{ij} has revenues and expenses that are summed up in an account Acc_{ij} of monetary units. Acc_{ij} constantly diminishes in the course of time.

It is relevant to mention that the monetary units that a module agent has on its account do not relate to the prices customers pay. The account only serves as an internal steering mechanism for a multi-agent system. The account surplus rather refers to the actual resources of an agent MA_{ij}. From definitions 2 and 3, we can conclude that each agent endeavors to maximize its account surplus. A surplus of zero will mean that the module agent risks death leading to the elimination of the module variant. Furthermore, the second part of definition 3 means that the agent's resources diminish in the course of time even if the agent does not carry out any task. To explain what a task is, we provide the following definition:

Definition 4: Module agent's task
The task T_{ij} of module agent is to form product variants by joining coalitions $C_k, k = 1,\ldots,n$.

The allocation of tasks to groups of agents is necessary when tasks cannot be performed by a single agent. The module agents on their own are not able to provide a solution. They need to cooperate in order to fulfill tasks. However, the autonomy principle of agents is preserved because each agent can decide whether to take part or not in a product variant. By forming coalitions each agent strives for its personal utility/account via cooperation. Module agents follow the economic principle of rationality and attempt to form a coalition which will maximize their own utilities. Furthermore, because of the heterogeneity of customer requirements, module agents may have different efficiencies in task performance due to their different capabilities.

In order to participate in a coalition, the module agent has to pay a certain fee. It is noteworthy that as opposed to other work in multi-agent systems (e.g. Shehory and Kraus, 1995), one agent may participate in more than one coalition. Moreover, these coalitions are dynamic and take place in real-time, after capturing customers' preferences. However, a coalition may succeed or fail. This primarily depends on the coalition's result, which can be complete or incomplete:

Definition 5: Complete vs. incomplete coalitions
We say that a coalition is complete if the coalition formed by the module agents builds up a salable product variant. A coalition is incomplete if the coalition formed by the module agents does not build up a salable product variant.

Note that an agent will join a coalition only if the payoff it will receive in the coalition is greater than, or at least equal to, what it can obtain by staying outside the coalition (Shehory and Kraus, 1995). In our case a module agent that does not participate in any coalition has a payoff of zero. Because the account surplus of an agent diminishes in the course of time, each agent should be interested in participating in beneficial coalitions to be able to reconstruct its resources and thus better ensuring its existence. However, there is no certainty about the success of coalition results. As aforementioned, each agent has to pay a certain fee in order to be allowed to participate in a coalition. But if the coalition that subsequently forms is incomplete or fails because it is not powerful enough to be displayed to customers, then the participation of an agent in a coalition is a waste of resources. Therefore, each agent has to be capable of estimating in advance the likelihood of the success of the coalitions it joins. However, the module agents should remain as simple as possible and should not become very complex. The whole multi-agent system has to contribute to problem solving and one agent should only dispose of the intelligence it requires in order to not waste computational resources.

3.2 The Negotiation Process

Firstly, it is relevant to determine the mechanism initiating a coalition. This being in reference to decisions about (a) which module agents are able to begin the formation of coalitions and (b) which reaching agreement process should be implemented to coordinate the module agents. We agree that platform agents are most suitable for initiating coalitions. We also assume that these agents dispose of an infinite account surplus. Therefore, they do not have to care about their existence. This is a legitimate assumption because the development of product platforms is generally cost-intensive. The development process itself may last for a duration of many years. Platforms are also created to be the basic module of a wide range of product variants for a long period of time. For example, by canceling a platform in the automobile industry, this would mean to cancel all models and the corresponding variants which are supported by this platform. Thus, such a decision is strategic and should not be allocated to automated software agents. As in each decision in variety steering it should be supported by human agents who have the required competencies and information. However, it is conceivable that each platform agent strives for being successful as much as possible, e.g. by contributing to the most sales' volumes.

On the basis of customers' preferences, the type and the number of product platforms to form coalitions are determined. Note that:

- a platform agent can be selected more than once,
- each product variant is based on one platform,
- each platform can be found in several product variants and,
- the total number of the selected platform agents is also the utmost limit of the product variants which can be formed by coalitions.

The coalitions take place at a certain point in time and form in order to fulfill the needs of one customer. When all resulting coalitions are complete, then the number of product variants will be exactly equal to the number of selected platform agents provided that no identical coalitions form. The platform agents have the ability to steer the formation of coalitions by (a) fixing the set of module agents which could contribute to the fulfillment of customers' requirements and by (b) determining the mechanism according to which module agents can join coalitions.

We propose to base the coalition formation mechanism on the target costing concept and a Dutch auction. Target costing is based on the price the customer is willing to pay. Starting from this price, it is possible to determine the utmost limit of the costs of each product function that is allowed to incur by taking into account the contribution of each function to the fulfillment of customer requirements. Further on, because each product component or module makes a certain contribution to the realization of a product function it is possible to distribute the function costs on the modules respectively components (Seidenschwarz, 2001). Thus, the result of target costing is an utmost limit for modules' or components' costs. The platform agents which are the auctioneers use these costs to initiate a Dutch auction. The module agents which compete to join the coalitions are the bidders. A Dutch auction is legitimate in this case because each agent tends to delay as much as possible in joining a sub-coalition in order to (a) better evaluate whether to participate or not in a hitherto sub-coalition and (b) minimize as much as possible the fees to pay. But due to the product configuration constraints, when a module agent wins the bid, it may impose constraints on the other bidding agents which intend to take part of the sub-coalition. Thus, the intelligence of the module agent should enable it to proficiently estimate when and for which coalition it bids. These auctions will continue until all coalitions are formed.

Although platform agents have an infinite account, we assume that they will also try to maximize the revenues they receive from module agents. We will describe in the next section how the product variants resulting from the coalitions are filtered after their formation. Only the set of product variants that are displayed to customers will receive revenues which are distributed on modules. The total collected monetary units from all module agents are collected in an account and then distributed on the module agents participating in the few successful product variants by considering their contribution in the fulfillment of the product functions and their participation level.

Up to now, we have only described what module and platform agents should perform and how the whole multi-agent system can reach agreements in order to form coalitions. But we have not mentioned what are the abilities an agent should have, to effectively carry out its tasks. Module and platform agents have different tasks. Therefore, they have different abilities. Module agents strive for maximizing their utilities (accounts). That is why, they have to develop strategies in order to survive. Subsequently, they should be able to evaluate in advance the success of the coalitions by estimating the probability that the formed product variants can be displayed to customers. Furthermore, they have to know when to bid and which coalition would be beneficial to join. Generally, as intelligent agents module agents have to update their knowledge from their own experience and the behavior of the other module agents pertaining to the multi-agent environment, which means that they have to learn.

In opposition to module agents, platform agents do not care about their existence due to their infinite account surplus. Furthermore, they decide which module agents are allowed to participate in the coalitions. Therefore, they are more powerful than module agents. Platform agents initiate and coordinate the formation of coalitions. They are also capable of communicating with each other to avoid the formation of identical coalitions. Platform agents have the overview of the coalition while forming and can forbid the further bidding of module agents by considering the constraints imposed by module agents which have already joined the coalition.

In the following we concentrate on module agents. We assume that the product platforms are capable of initiating the Dutch auction and that only product constraints may restrain the participation of module agents in coalitions. In order to represent the module agents, we use a mathematical tool from decision theory. Decision theory defines a rational agent as one that maximizes the expected utility. The expected utility EU of an action is defined as (Russel/Norvig, 1995):

$$EU(\alpha) = \sum_{\omega \in \Omega} U(\omega)\, P(\omega \mid \alpha)$$

where

$\alpha \in Ac$; Ac: the set of all possible actions available to an agent,

$\Omega = \{\omega, \omega', \ldots\}$: the set of all possible outcomes

$U : \Omega \to IR$ a utility function associating an outcome with a real

P: a probability distribution over possible outcomes given the performance of actions

Let the function f_{opt} take as input a set of possible actions, a set of outcomes, a probability distribution and a utility function and let this function return an action. The defined behavior of f_{opt} is defined as follows (Russel/Norvig, 1995):

$$f_{opt}(Ac, \Omega, P, U) = \arg\max_{\alpha \in \mathrm{Ac}} \sum_{\omega \in \Omega} U(\omega) P(\omega \mid \alpha)$$

Wooldridge (2000) criticizes f_{opt} for building rational agents because f_{opt} requires an unconstrained search which can be very expensive when the space of all actions and their outcomes is very wide. But, in our case this critic does not seem to be strong enough because the action types that a module agent can perform are (a) to participate in a coalition (Participating is the action $\alpha = 1$) or (b) not to participate (Not participating is the action $\alpha = 0$). Thus, the number of action types a module agent disposes of are only two. Furthermore, the outcome of actions may be that either (a) the module agent is a member of a product variant which is selected in the final subset (Success of a coalition is the outcome $\omega = 1$) or (b) the module agent is a member of a product variant which is not selected in the subset to be presented to customers (Failure of a coalition is the outcome $\omega = 0$). That is why, we argue that it is legitimate to consider f_{opt} to build module agents. However, f_{opt} should be adapted to the requirements of the multi-agent system problem that was presented above.

Suppose at a point in time t=0 the platform agents initiate coalitions. Each platform agent chooses which module agents are allowed to bid. For each module agent, the platform agent communicates a function to a module class *i* having the following form:

$$K_i(t) = K_i g_i(t)$$

where

$K_i(t)$: the Dutch auction function decreasing in the course of time,

K_i : a constant representing the first price of the Dutch auction,

$g_i : [0,T] \to]0,1] / \, g_i(0) = 1;$

g_i : a steadily decreasing function and,

T : the maximal bidding time

As aforementioned, when a module agent joins a coalition, it may restrain the participation of other module agents also intending to join the coalition. Therefore, each module agent must be capable of evaluating the behavior of the other agents that could prohibit its participation. This behavior should be captured by the function $Risk$ which is a risk function of a module agent MA_{ij} :

$$Risk : [0,T] \to [0,1] / \, Risk(0) = 0 \text{ and } Risk(T) = R$$

where

$Risk$ is a steadily increaing function

$R \in [0,1]$ is a constant reflecting the risk willingness of the module agent

Note that the function $Risk$ is a function that leads the module agent to bid as early as possible in order to increase its chances in being a participant of a coalition. Let $\mathrm{Re}\,venue$ be the function which takes the value 0 when the agent is not a member of the coalitions representing the product variants displayed to customers and the value $\mathrm{Re}\,v$ when the product variants are displayed to customers.

The utility function U of a module agent depends on the risk function which is supposed to decrease revenue during the auction process, the revenue function and the Dutch auction function. The adapted f_{opt} for our case is:

$$f_{opt}(Ac, \Omega, Risk, \mathrm{Re}\,venue, g_i, P, U) = \arg\max_{t\in[0,T]} \sum_{\omega\in\{0,1\}} \{[1 - Risk(t)]\mathrm{Re}\,v - K_i g_i(t)\} P(\omega \mid \alpha)$$

The adapted f_{opt} returns the point in time t at which the module agent has to bid for the Dutch auction to maximize its utility. But note that if $t \in [0,T]$ then $\alpha = 1$ and if there is no $t \in [0,T]$ maximizing the utility ($t > T$) then $\alpha = 0$ and the module agent intends on not participating in the coalition.

Furthermore, suppose that a module agent MA_{ij} is allowed to participate in p coalitions: $C_k, k \in \{1,\dots,p\}$. For each coalition the module agent estimates f_{opt} and at the point in time t=0 where an auction begins, the module agent has to develop a plan

$$(Plan_{t=0})_{ij} = \left((\alpha_1^0, t_1^0), \dots, (\alpha_k^0, t_k^0) \dots (\alpha_p^0, t_p^0)\right)_{ij}$$

which indicates whether or not and when to bid for each coalition. For notation purposes, when $\alpha_k = 0$, the module agent allocates to t_k^0 an infinite value $(t_k^0 = \infty)$, which is in accordance with the fact that an agent will never bid and then not to participate. Moreover, by developing a plan the module agent has to consider its account constraint. It is not allowed to pay for coalitions more than the account surplus. This means that $\sum fees \leq Account\ surplus$. It is also conceivable that the module agent wants to allocate only a certain sum of monetary units for the coalitions which should be formed to be presented to one customer. This depends on the self-preservation strategy the module wants to pursue.

However, the agent plan determined at $t = 0$ is not fixed for the whole auction process. The module agent has to adapt this plan according to the changes which could occur in its environment. Suppose that the tuples of $(Plan_{t=0})_{ij}$ are arranged so that $t_1^0 \leq \dots \leq t_k^0 \leq \dots \leq t_p^0$. Suppose that $t_1^0 \neq \infty$ *and* $t_2^0 \neq \infty$ and that at a point in time $t < t_1^0$ an agent from the same class wins the bid or an agent from another class imposes participation constraints. At this point in time, the module agent has to estimate once again f_{opt} for the remaining coalitions to determine whether and when to bid. This is legitimate because when the participation in one coalition fails the module agent can allocate the resources he has planed to expend differently. The resulting plan at a point in time $t = 1$ is:

$$(Plan_{t=1})_{ij} = \left((\alpha_2^1, t_2^1), \dots, (\alpha_k^1, t_k^1) \dots (\alpha_p^1, t_p^1)\right)_{ij}.$$

The application of the described process will continue until different coalitions are formed.

Recapitulating, we can say that the main advantages of the developed multi-agent approach are:

- the easy maintenance of the system: when introducing new module variants or eliminating old ones, it is sufficient to introduce or eliminate module agents,
- the dynamic variety generation during the interaction process and variety steering as well as,
- the application of a market mechanism concept which lets the intelligent agents themselves decide according to the current situation about their suitability to fulfill real customers' requirements. Such a market mechanism based approach enables us to efficiently carry out the coordination mechanism, even for a high number of involved agents (Shehory et al., 1998).

4 TECHNICAL ARCHITECTURE

In this section we present a complete model for variety formation and steering based on the multi-agent system approach developed in the previous section. We propose to interface the module agents' pool to a customer advisory system to support dynamic variety formation during the real time customer advisory process.

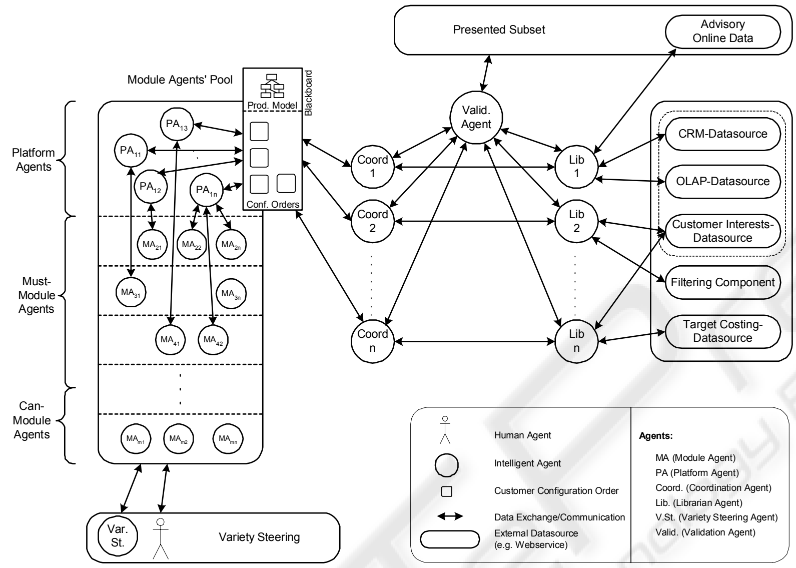

Figure 1: Technical architecture for an agent based variety formation and steering

Advisory systems are software systems that guide customers according to their profile and requirements through a „personal", customer oriented advisory process to elicit their real needs from an objective point of view (Blecker et al., 2004). During the advisory process, the system suggests the customer product variants according to his profile and refines them through the dialog. At the end of the advisory process, the customer is supported with product variants which fulfill his real needs.

At each advisory session the multi-agent system dynamically creates coalitions of product variants that can be recommended to the user. Therefore, we aim at integrating the system into the existing information system landscape. Figure 1 depicts the architecture of such a system.

Beside the agents' pool the architecture consists of the following main elements:

- an online interface to the data of the advisory system that provides a customer's preferences,
- an interface to the supplier's back office which for instance comprises a CRM or OLAP system,
- additional filtering and target costing data sources,
- librarian agents that have access to all back office systems and make proper data available for the other components,
- coordination agents that coordinate the variety formation in the agents' pool and,
- a blackboard that supports communication between the module agents' pool and its environment.

The system also supports variety steering. As was mentioned in the previous section, the account balance of the agents provides a measurement of the success of a module variant which constantly can be analyzed by a variety steering agent or humans.

Before we technically describe the system, we will describe the main idea on the basis of a scenario: During the advisory process the system captures the customer's requirements according to his knowledge level. During the advisory dialog the system presents the user a proposal of several product variants according to his profile and preferences. These are refined through the advisory dialog which leads to dynamically refined suggestions for product variants. Finally, the system generates suggestions of product variants that meet real customer needs.

The creation of a valid subset of product variant coalitions is dynamically carried out by following steps:

(1) A so-called librarian agent obtains data from the online advisory data source. These data can contain both user data and product preferences – de-

pending on the knowledge level of the user. If the user is familiar to the domain, he can make a decision on the product level; otherwise the system gathers his needs, which can be captured in e.g. a language different form product specification. For instance, data can contain personal data such as the customer's age or marital status, his personal preferences or desired product attributes. In the automotive domain it could be about a male customer with two children who is sporty, but prefers an economical car.

(2) The information about the customer is supplemented by the librarian agent: Depending on whether the customer is recognized (e.g. by a login process) this data can be obtained from the CRM data source where both the customer's interests and his past buying behavior are stored. Otherwise the information can be provided from the OLAP data source where traditional web mining techniques such as clustering are used to extend the customer's profile. The result of this process is an improved profile of the customer's needs.

(3) In order to support the negotiation process in the module agents' pool, the librarian agent calls for service from the filtering component in order to convert the customer attributes to product attributes. For instance, this can be based on expert rules or statistical methods. As an example the attributes of (1) could be inferred that the car should be a sedan with no automatic transmission that runs on diesel.

(4) The target costing component is used to estimate the costs of the product's functions that the customer probably will have to pay for. For instance, this could be based on past sales of a clustered customer group.

(5) The data is passed on to the coordination agent who monitors the load of the module agents' pool. If the number of customer configuration orders is below a certain limit, the coordination agent sets forth a request for new product variants with the desired product properties onto the blackboard. Note that these product properties derived from the customer attributes only support the negotiation process, they are not constraints. Besides for that, the coordination agent selects both appropriate platform agents who should carry out the auction, and the number of coalitions of product variants they should form. This decision is based on the customer's profile and the product model.

(6) Now the negotiation process is carried out as described in the previous chapter, until all coalitions are formed. The resulting coalitions are put back onto the blackboard where they are removed from the coordination agents and passed to the validation agent.

(7) The validation agent requests data from information agents in order choose a subset of the available coalitions of variants to present them to the customer. This task is performed on the basis of validation which is a kind of „reverse mapping". That means that the properties of the selected coalitions are mapped to customer attributes as a kind of verification. The best coalitions are presented and rewarded with monetary revenue for the accounts.

(8) If the customer makes the decision to buy a certain product variant of the presented subset, it is conceivable that an additional reward would be sent back to the accounts of the corresponding module agents.

(9) Additionally, we propose the use of the account level of each module agent as an indicator to support variety steering decisions. In an independent process the system makes suggestions to eliminate module variants. If the account level is negative or low in comparison with competing module variants, this is an indicator to remove the module variant. Furthermore, the introduction of new module variants can affect the internal and external complexities which can be estimated by computing suitable key metrics (Blecker et al., 2003).

In conclusion we can see that the complexity is spread throughout all system components in the multi agent system:

- The module agents' pool is responsible for carrying out a negotiation for forming product variant coalitions,
- coordination agents manage the blackboard and the general interface between the agents' pool and its environment,
- librarian agents not only interface the back office systems, they independently obtain data and process them in an intelligent way to support the other agents optimally,
- a validation agent carries out the validation of the module variants coalition independently of the decisions of the other system components.

For the implementation of such, we propose to base the system on Java technology. This not only ensures platform independence, it also provides a uniform framework for the implementation of all components.

All back office systems such as CRM, OLAP or other data sources must be connected via custom interfaces, for example by XML. On the variety formation system, data can be provided by web services so that the agents can access the services. The agents' pool can be realized in one virtual machine. Due to the decision to represent module variants instead of product variants as agents, this assumption is admissible. This way we can lower the communication costs because this enables a direct interaction between the agent instances. The coordination between the agents' pool and the external agents is carried out via a blackboard where all agents are registered. Coordination agents, validation agents and librarian agents can be distributed for reasons of load balancing. Communication between these agents can be performed via Java's RMI (Remote Method Invocation) or CORBA to support other systems.

5 CONCLUSIONS

In this paper, we have depicted the main problems which are triggered by increasing variety in mass customization. Variety involves an internal complexity inside a company's operations, as well as an external complexity from a customer's perspective. To mitigate both complexities' problems, the main idea is to provide an information system solution which is capable of both supporting customers during the interaction process by proposing and refining product variants and simultaneously supporting variety steering decisions. The agent technology is able to be identified as a suitable approach to cope with this problem in a decentralized, self-coordinating way.

The developed system integrates both customer's and supplier's perspectives in one information system. We outlined how module variants can be represented as intelligent agents that negotiate with each other to ensure their survival within the scope of variety steering. Based on the decision theory's model for rational agents, we formally define the function that an agent strives to optimize. The negotiation process between the intelligent agents is based on the target costing concept and a Dutch auction. This is also described in a formal way defining the possible functions which have to be determined. Because we intend to carry out simulations of the entire system, several functions which determine the intelligence of the defined agents should be tested. Based on these simulations we will decide which implementation will lead to a working prototype. Furthermore, a technical architecture for the agent-based variety formation and steering in mass customization is proposed.

The main advantages of the developed approach are the easy maintenance of the system, the dynamic variety generation and variety steering, as well as the application of a market mechanism concept supported by agent technology. The adopted market mechanism presents a relevant approach enabling one to overcome the shortcomings of existing interaction systems and variety steering methods. Thus, instead of building rigid rules in the interaction system that map customer requirements into product attributes, the proposed market mechanism approach lets the intelligent agents themselves decide according to the current situation about their suitability to fulfill real customers' requirements. Furthermore, the market mechanism enables us to connect two relevant concepts in mass customization, namely which product variants should be retained in the product assortment and which specific ones from this assortment should be selected and offered to a particular customer.

REFERENCES

Blecker, T., Abdelkafi, N., Kaluza, B., and Friedrich, G., 2003. Key Metrics System for Variety Steering in Mass Customization. In *MCPC'03, 2nd Interdisciplinary World Congress on Mass Customization and Personalization, Munich*, October 6-8, 2003.

Blecker, T., Abdelkafi, N., Kreutler, G., and Friedrich, G., 2004. An Advisory System for Customers' Objective Needs Elicitation in Mass Customization. In *4th Workshop on Information Systems for Mass Customization, Madeira Island,* February 29-March 3, 2004.

Ericsson, A., and Erixon, G., 1999. *Controlling Design Variants: Modular Product Platforms*, Society of Manufacturing Engineers. Dearborn, Michigan.

Iyenger, S. S., and Lepper, M. R., 2000. When Choice is Demotivating: Can One Desire Too Much of a Good Thing?, URL: http://www.columbia.edu/~ss957/publications.html (Retrieval 10. Oct. 2003).

Kauffman, S. A., 1993. *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York.

Lingnau, V., 1994. *Variantenmanagement: Produktionsplanung im Rahmen einer Produktdifferenzierungsstrategie*, Erich Schmidt Verlag GmbH & Co. Berlin.

Piller, F. T., 2001. *Mass Customization: Ein wettbewerbsstrategisches Konzept im Informationszeitalter, Gabler Verlag*. Wiesbaden, 2nd edition.

Piller, F. T., and Waringer, D., 1999. *Modularisierung in der Automobilindustrie – neue Formen und Prinzipien*, Shaker Verlag. Aachen.

Pine II, J., 1993. *Mass Customization: The New Frontier in Business Competition*, Harvard Business School Press. Boston.

Rathnow, P. J., 1993. *Integriertes Variantenmanagement: Bestimmung, Realisierung und Sicherung der optimalen Produktvielfalt*, Vandenhoeck und Ruprecht. Goettingen.

Rautenstrauch, C., Taggermann, H., and Turowski, K., 2002. Manufacturing Planning and Control Content Management in Virtual Enterprises Pursuing Mass Customization. In Rautenstrauch, C., Seelmann-Eggebert, R., and Turowski, K. (Eds.). *Moving into Mass Customization: Information Systems and Management Principles*, Springer-Verlag. Berlin Heidelberg, pp. 103-118.

Rosenberg, O., 1997. Kostensenkung durch Komplexitaetsmanagement. In Franz, K.-P., and Kajueter, P. (Eds.). *Kostenmanagement: Wertsteigerung durch systematische Kostensenkung*, Schaeffer Poeschel Verlag. Stuttgart, pp. 225-245.

Russel, S., and Norvig, P., 1995. *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Inc. New Jersey.

Schwartz, B., 2000. Self-Determination: The Tyranny of Freedom, URL: http://www.swarthmore.edu/SocSci/bschwar1/self-determination.pdf

Seidenschwarz, W., 2001. *Target Costing. Marktorientiertes Zielkostenmanagement*, Vahlen. Muenchen, 2nd edition.

Shehory, O., and Kraus, S., 1995. Coalition formation among autonomous agents: Strategies and complexity. In Castelfranchi, C., and Muller, J. P., (Eds.). *From Reaction to Cognition*, Springer Verlag. pp. 57-72.

Shehory, O., Sycara, K.P., and Jha, S., 1998. Multi-agent coordination through coalition formation. In Singh, M.P., Rao, A.S., and Wooldridge, M.J., (Eds.). *Proceedings of ATAL'97*, Springer Verlag. pp 143-154.

Wildemann, H., 2001. *Das Just-In-Time Konzept – Produktion und Zulieferung auf Abruf*, TCW Transfer-Centrum GmbH. Muenchen, 5th edition.

Wildemann, H., 2003. *Produktordnungssysteme: Leitfaden zur Standardisierung und Individualisierung des Produktprogramms durch intelligente Plattformstrategien*, TCW Transfer-Centrum GmbH. Muenchen, 2nd edition.

Wooldrige, M., 2000. *Reasoning about rational agents*, The MIT Press. Cambridge et al.