ONTOLOGIES SUPPORTING COOPERATIONS IN MASS CUSTOMIZATION – A PRAGMATIC APPROACH

Dirk Pawlaszczyk, Technische Universität Ilmenau, 98693 Ilmenau, Germany
dirk.pawlaszczyk@tu-ilmenau.de

Andreas J. Dietrich, Universität Hohenheim, 70593 Stuttgart, Germany
adietr@uni-hohenheim.de

Ingo J. Timm, University of Bremen, 28334 Bremen, Germany
itimm@acm.org

Stephan Otto, Universität Hohenheim, 70593 Stuttgart, Germany
ottosn@uni-hohenheim.de

Stefan Kirn, Universität Hohenheim, 70593 Stuttgart, Germany
kirn@uni-hohenheim.de

ABSTRACT

Mass Customization (MC) as a modern competitive strategy enables enterprises to act in dynamic markets with high customer satisfaction. But, for a successful implementation of this management concept, effective information logistics is essential. Ontologies are considered as a promising approach for optimizing inter-organizational and distributed cooperation.

Although there have been several publications on ontologies and their usage we are going to show that none of these approaches can satisfy the requirements of the Mass Customization domain completely. They either do not consider requirements like natural distribution and inherent heterogeneity of all members along the supply chain nor customer needs sufficiently, which are both typical properties in this domain.

In this paper, a formal conceptualization of Mass Customization scenarios within an ontology will be introduced. Doing so, we will firstly illustrate the general concept of ontologies. Then we will discuss the components of an MC-ontology and its sub-models. The originalities of a particular domain are shown for the shoe industry as an example.

KEYWORDS

Mass Customization, ontology, multiagent system, inter-organizational cooperation

1. INTRODUCTION

During the last years, economy is changing from an enterprise driven to a buyer’s market. Consumers have precise wishes about the goods they want to buy and enterprises must offer a diverse product portfolio in order to satisfy these customer’ needs. As holistic and large approaches are necessary for getting ready and being successful within this rough situation, several so-called competitive strategies have been developed (e.g. cost leadership by Michael Porter (Porter, 1999)). Mass Customization (MC) is one of such innovative management concepts. By combining individuality of customized goods with adequate low prices as known from mass production, this concept has been considered to reach high customer’s satisfaction. Thus, flexibility and low response times are requirements for the desired variety and customer orientation in businesses (Pine, 1993; Piller, 2003).

Within the joint research project “EwoMacs”, logistics structures of MC in the shoe industry are analyzed. EwoMacs aims at developing business models for optimized logistics in MC. The project consortium consists of three research and five industrial partners and is funded by German Federal Ministry for Education and Research (BMBF, 02PD1120). As part of the project, a simulation system will be developed for analyzing, simulating, and evaluating information as well as material flows. The

University of Hohenheim serves as a technological partner and is conducting research on simulating the corresponding logistics structure of the industry partners with an agent-based approach. Ontologies are considered as a promising approach for interaction in highly distributed systems, such that it is of high interest, if ontologies can be used for optimizing inter-organizational and distributed cooperation. In this paper, we are introducing MC-adapted ontology that has been created within EwoMacs.
In section 2 we are introducing to MC, semantic interoperability, and ontologies. In the following section, coordination and cooperation problems within MC as challenges for enterprises are specified. We present both generic and domain-specific MC-ontologies in section 4. Further on, we consider an application scenario of the ontologies. In order to compare our approach with existing research related work is discussed in section 5. Finally we are summarizing the benefits and shortcomings of the approach and provide an outlook for future work.

2. RATIONALE

The foundations of this paper can be found in three main research fields: Mass Customization, semantic interoperability, and ontologies. The first paragraph of this section will introduce the general concept of MC with specific requirements of its logistics. In the second paragraph, the problem of interaction and possible solutions of semantic-based interaction are discussed. Conclusively, the concept of ontologies is presented in detail.

2.1 Mass Customization

As described above MC is a modern management strategy. This concept is best defined as a transaction process, which focuses on individualization of mass-market products and services to satisfy specific needs of the customer, at an affordable and reasonable price. Therefore, MC could satisfy customer's current needs: regularly new and individual products and low prices. Thus, enterprises have to manage short product life cycles and a high degree of flexibility within the production process as well as to take advantage of efficiency potentials (for example economies of scale or economies of scope). Theoretically, there is no restriction with respect to the area of industry MC could be applied to. In order to organize the diversity of this strategy several classification schemes have been developed and published (e.g. Gilmore/Pine, 1997; Piller, 2003). They illustrate explicitly that MC could be implemented to different types of goods (product or service) and branches (automotive industry, shoe industry etc.). Understanding MC as an abstract business model, it has impacts on product design, manufacturing, and assembly processes as well as logistics and information processing, e.g. small lot sizes and increased diversity of variants. Four main issues should be considered (cp. Piller/Stotko, 2002):

- Split of production process,
- Flexible production control due to ex ante loose specification of products,
- Information on individualization throughout the value chain,
- Unique identification of each product.

During the production process of mass customized goods, the so-called decoupling point is indicating the point within the supply chain, which differentiates customized production from mass production. I.e. after that point each former anonymous order is assigned to a specific customer. MC is characterized by defining the decoupling point after receiving the customer specific order. Individual configurations are created with regard to width and depth of variation. However, underlying production control processes, e.g. production control, planning, and scheduling, have to realize a high degree of flexibility, ensuring robust production processes – even in logistics networks – on the basis of ex ante imperfect product specifications. In consequence, production processes should be defined independently from specific configuration parameters. Furthermore, information of customized orders has to be propagated throughout the network as soon as possible.

2.2 Semantic Interoperability

Efficient information management and the processes therein become more and more important within MC. Two main research questions arise here: how to manage the process of interoperability and how to interpret the content of exchanged information. In Distributed Artificial Intelligence, there are approaches under research, i.e. Intelligent Agents, which are dealing with intelligent interaction of autonomous decision makers. The interest in agent technology and agent related topics have risen enormously in the last decade. Intelligent Agents can act autonomously, communicate with other agents, are goal-oriented (pro-active) and are using explicit knowledge (Weiss, 1999). An adequate application of Intelligent Agents should meet the following three criteria (Müller, 1997): A natural distribution of the participating entities (e.g. resources at the shop floor), a dynamic environment where structures and conditions are continually changing, and complex interaction and co-ordination between the individual entities. Since most settings in MC meet these requirements, the use of an agent-based approach with autonomously, co-operatively and purposefully acting intelligent software units seems to be suitable. Communication within agents is one major aspect of this technology. The agent communication language is defining process, syntax, and content of communication. The approach of agent-based interaction is based on defining a conversation as a structural element of communication processes, which consists of a sequence of messages (communicative acts). There are different interaction protocols available for selecting messages within a conversation (FIPA, 2002). The interaction protocol is directly connected to the aim of communicating. I.e. there are auction protocols provided by an agent communication language, which are used for price negotiations. Until recently, many agent applications have remained nothing more than small pilot projects in the research laboratory. One key reason behind this is that the necessary network and communication infrastructure for agent deployment lacked robustness, openness and interoperability to support heterogeneity within a hybrid agent – non-agent world. Nowadays, this is changing as ongoing standardization efforts within bodies such as FIPA, the Foundation for Intelligent Physical Agents, and the W3C are providing standard interaction mechanisms for agent based software and Web technologies. FIPA’s aim is to promote technologies and interoperability specifications that facilitate the end-to-end internetworking of intelligent agent systems in modern commercial and industrial settings. During the last few years, FIPA's agent communication standards have been gaining the most traction amongst vendors and the user community. Thus, agent technology may be used for managing interaction between distributed processes within MC.

The second problem within semantic interoperability is, that the content of messages has to be connected to context, such that each agent will use the same interpretation of information shared within the network. The information used will be part of business information systems, enterprise resource planning systems, etc. Most information systems use specific data models and databases for this purpose. This implies that making new data available to the system requires, that the data must be transferred, into the system's specific data format. This is a process, which is very time consuming and tedious. Data acquisition, automatically or semi-automatically, often makes large-scale investment in technical infrastructure and/or manpower inevitable. These obstacles are some of the reasons behind the concept of information integration. Problems that might arise due to heterogeneity of the data are already well known within the distributed database systems community (e.g. Kim/Seo, 1991). In general, heterogeneity problems can be divided into three categories:

- **Syntax** (e.g. data format heterogeneity)
- **Structure** (e.g. homonyms, synonyms or different attributes in database tables)
- **Semantic** (e.g. intended meaning of terms in a special context or application)

For information management problems on the structural and semantic level with regards to terminologies are important. Terminologies are important because they contain the companies' knowledge. The IT manager is confronted with the task of how to map one terminology to another terminology. Lately, approaches based on formal ontologies have shown that they are promising. In current research of this field, there are approaches for ontology-based intelligent information integration (Wache, 2003). The underlying assumption is the use of an ontology for specifying meta-data. The concept ontology will be discussed in the next paragraphs.

2.3 Ontologies

For better understanding of the following discussion it is inevitable to make some short remarks concerning terms like “conceptualization” and “ontology”. Furthermore the current section is introducing a probably useful classification scheme that exposes different types of ontologies to the reader. Finally some principles for engineering and design of ontologies as well as some application fields will be described.

Definition

The term “ontology” is associated with different meanings within particular disciplines. In computer science, it is normally apprehended as a formal, explicit specification of a conceptualization (Gruber, 1993). An ontology can be used to describe entities and their properties in a formal manner. Therefore it covers all objects of interest, together with definitions for the meaning of each of the terms in a given domain. Formal axioms are used to enforce constraints on the entities to describe their behavior.

More precise the ontology represents the intended meaning of a formal vocabulary related to a particular conceptualization of the world (Guarino, 1998). In an order process for example, concepts like “quotes” and “offers”, although part of different ontologies, can have the same intended meaning (they are both meant to represent a concrete proposal in the given context). Consistently you can use different vocabularies on top of the same conceptualization. Thus, the ontology offers a language-dependent description of our universe of discourse, whereas the conceptualization itself is language-independent. Despite this high level of specification ontologies offer flexibility, by sharing and reuse of ontologies and the ability to accommodate varying descriptive terms.

Ontology types

Depending on their level of generality ontologies can be discerned into different types (Guarino, 1998; Becker et al., 2003). Common concepts like “time”, “action”, or “process” that are normally not associated to a particular problem domain, but rather are universally applicable, can be subsumed in so called “Top-Level” or “General” ontologies.

A “domain ontology” covers the semantics and vocabulary related to a generic domain (like jurisdiction, automotive or even MC). Therefore it unifies higher level concepts representative to a set of application scenarios. Generic tasks or activities like selling or manufacturing on the other hand can be represented inside of a “task ontology” (Guarino 1998), whose scope of generalization is comparable to the domain ontologies.

While the last two ontology types are on the same level of abstraction, the remaining “application ontology” type is meant as a refinement to them. Representatives of this class depend on a particular domain and task as well. Concepts of an application ontology often correspond to roles of entities which perform (or possibly get used by) a certain activity, like cutting leather or forwarding commodities. Thus, they combine static as well as dynamic aspects.

Ontology Design

Before starting with the actual design process, it is necessary to define the ontology’s purpose together with the field of application it is meant for. Further on you need to choose a description technique that fits to your needs, because codification can be done in many different ways. To describe concepts and relations of a problem domain you can use simple text phrases. This might be sufficient especially in situations were you need consensus concerning some terms, but won’t go further than that. If we want to support automated information processing, a more formal design approach is needed. Most common formalisms used to represent ontologies are descriptions logics like KL-ONE (Brachman/Schmolze, 1985) and the Knowledge Interchange Format (KIF, 1998), a language based on first order predicate logic that enables support for representing meta-knowledge. Unlike a pure textual representation scheme such formal languages provide a much stronger axiomatization and beyond this, they support deduction capabilities as well. In recent times also object-oriented modeling techniques like the Unified Modeling Language (UML) with its standard graphical representation models gain more and more popularity in ontology design (Cranefield/Purvis, 1999). This design approach is commonly denoted to be semi-formal, because there is no support of deductive capabilities like in KIF or KL-ONE. But on the other hand, for pure integration of distributed information systems for example, such feature is not necessarily needed. However, there is always a trade off between expressive power and tractability when using a formal language to represent your knowledge.

With this in mind, the actual modeling process can start. Modeling normally begins with identification of concepts and terms, specific to our domain of interest. For each term we have to specify its meaning in the given context. Furthermore we need model concepts, relations and possible axioms by using predefined constructs of our description language. The concrete development process normally starts with a small model that is completed in the following refinement steps. There is not just one correct way for ontology building (Jones et al., 1998). Actually we can discern several approaches for ontology design process, depending on the number of ontology engineers and the starting point of the particular project, whereby concrete proceeding guidelines are suggested (Holsapple/Joshi, 2002). We can distinguish between inductive and deductive design approaches as well as inspirational (single person) and collaborative (many persons) ones. Furthermore, (Gruber, 1993) outlines some design criteria to guide the ontology development such as clarity, coherence, extendibility or minimal ontological commitment.
Finally, the ontology should be tested with a representative scenario, i.e. by means of the use case it was designed for to reveal possible redesign efforts.

Application areas
Ontologies can be used to promote re-use and sharing of knowledge. They are intended to enable support for communication, interoperability as well as systems engineering tasks like requirements specification or re-usability (Uschold/Gruninger, 1996). Because of their expressivity to define terminologies used in a domain or in an application they are a key technology for information integration in various application fields, such as electronic commerce (Ontoweb, 2002), enterprise application integration (Gruninger/Fox, 1996), supply chain management (Vikram, 2003), and knowledge management (Smirnov/Chandra, 2000). The notion of ontologies also plays a central role in the emerging Semantic Web. The Semantic Web consists of Internet sites that provide information in a “meaningful” format for machines.

3. COOPERATION WITHIN MASS CUSTOMIZATION

MC is a new challenge for enterprises. The complexity of modern products may overextend the available skills, knowledge, and capacities of a single enterprise or it may force this enterprise to spend great efforts in research and development activities to meet the customer’ requirements. The problem of overextension is well known in global economy and in consequence enterprises are establishing cooperation to manufacture complex industrial products in a distributed fashion. The implementation of efficient consumer response within logistics networks enables MC in a very flexible manner. In this section we will introduce the specific requirements of collaborative MC for modern information management systems.

3.1 Collaborative Mass Customization

The focus in terms of MC in this paper lays on effects of distributed value chains as well as importance of information management. Firstly, MC scenarios in general consist of several, independent actors. For example, a vendor offers MC products at the market, a retailer provides the configuration process, a network of contractually joined enterprises are assembling or manufacturing the product and a forwarder completes the order by delivering it to the customer. A detailed illustration of typical actors within a MC scenario is shown in figure 1.

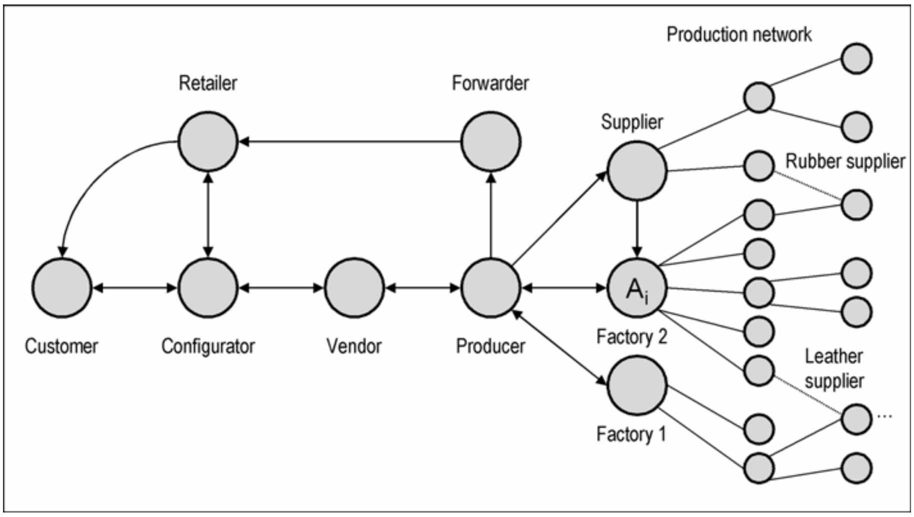

Figure 1: Typical actors in a MC supply web (based on Dietrich et al., 2003a)

Beneath difficulties of distributed production and logistics, the management of MC orders throughout the value chain is one of the most relevant efforts. Insufficient coordination between actors, adversely exploited autonomy of each actor and failing in information transmission are severe problems for smooth configuration, production, and delivery of MC products (cp. Sugumaran et al., 2003). Secondly, an essential target for MC models is to optimize information management. The importance of information in the field of this concept is obvious: The customer’s requirements have to be integrated in a product specification. In order to realize efficient and effective production of individual products, it is necessary to supply each actor of the value chain with information about the product and customer. Information systems can be used for efficient information management. Therefore, computer and information systems support is one success factor for implementing MC (Mertens, 1995; Piller, 2003).

3.2 Computer-mediated collaborative Mass Customization

Collaborative MC is a new challenge for supporting information systems. There are different types of systems, e.g. production planning, process control, enterprise resource planning, involved into the process of customizing, producing, delivering, and selling a MC product. If the product is built within a logistical network the complexity is increasing. In the collaborative MC case, there is need for integration the involved information systems – at least virtually. Here, problems of data privacy and security arise when two or more independent companies are interconnected. Only uncritical data needed for the common processes should be exchanged. In addition, owing to heterogeneous information systems, there is a problem in automatic negotiation within industry-wide co-operation relations. This problem is enforced by missing or inadequate standardizations for data exchange and various, partially contradictory definitions of used concepts. To address this problem, we propose an Intelligent Agent approach. This approach is based on agents, which are representing enterprises or profit centers within enterprises for the automated co-operation in logistic networks. Agent representing entire enterprises are usually modeled and realized as multiagent systems (cp. hierarchical agent systems). The multiagent system is providing a framework for co-operation within short-term relationships as needed for temporal logistics networks. So, our focus lays on sophisticated coordination skills like complex negotiation abilities and adaptive social behavior.

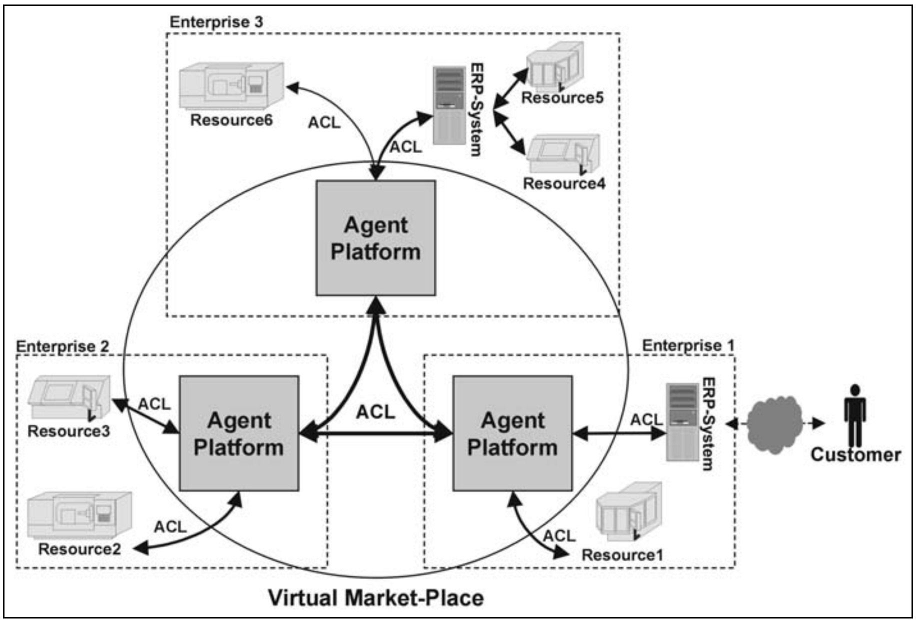

Figure 2: Multiagent System Architecture

These agents are linked to enterprise resource planning (ERP) systems. They manage information transfer partially automatically using semantically well-defined communication. An essential step for the success of this approach in the framework of real industrial scenarios is to provide an open specification of the agents. Thus, the system architecture for agent-mediated MC is based on the standardization efforts of the Foundation for Intelligent Physical Agents (FIPA) committee (FIPA, 2000). FIPA defines the crucial elements of agents, agent systems, and agent platforms prohibiting the establishment of new and mutual incompatible systems. On a conceptual basis the project will be realizing a new cooperation within MC based on an electronic market place, where each participating partner is handled identically. Even resources or ERP systems can be integrated in this market-place directly. The application of the FIPA reference architecture to this conceptual model will lead to a design of an agent system as illustrated in Figure 2. The basic concept of this design is that each enterprise is represented by an agent platform. This agent platform will be connected to each agent platform within the system using the agent communication language (ACL). The enterprises can decide on their own, which resources will be integrated within the platform. Consequently, if an enterprise uses an efficient ERP system, this system can be linked to the agent platform using an Intelligent Agent. This enables a planning and scheduling on the basis of the ERP system as well as a partner matching and negotiating process on the basis of complex interaction using an ACL.

Figure 2 demonstrates the advantages of this approach in a straight forward way: agents are the interfaces of the possibly heterogeneous system architecture. Therefore, composing large systems in this way is just a matter of the communication of the single agents. There a no other dependencies between parts of different enterprises and hence such a system is inherently modular, flexible, and extensible.

The application of cooperative multiagent system and intelligent agents seems to be very promising but is also opening up risks for the safety and security of enterprises and the robustness of the (distributed) production. The security issues associated with agents fall into three major groups: integrity attacks, privacy attacks, and denial of service attacks. A malicious agent may try to modify or delete information in the environment in unauthorized ways. The second form of attack consists of information theft or leakage: a hostile agent may try to get internal information from a cooperating enterprise. The third form of attack consists of denial of service, where the agent attempts to interfere with the normal operation (e.g. production process) of an enterprise. Therefore it is necessary to establish adequate security mechanisms or restrict the multiagent system to trusted cooperation partners only. Recent research deals with the first issue, but for implementation of multiagent systems in today's production processes, it seems to be necessary to apply the latter security approach and to restrict multiagent systems to trustful participants, which commitment is accompanied by external contracts.

In the last paragraphs, we proposed an agent architecture to manage the interaction within MC as introduced in (Timm et al., 2001). This approach is part of research within the "IntaPS" project, which is focused on integrating process planning and production control within a single enterprises. As a next step the Agent.Enterprise approach is integrating heterogeneous multiagent systems. The Agent.Enterprise approach is funded within the priority research program of the Deutsche Forschungsgemeinschaft on "Intelligent Softwareagents in Business Applications" (coordinated by Prof. Stefan Kirn). As a result, it was identified, that the main problem of interoperability is not the process of interaction, which can be solved by standardized protocols, but the adequate definition of an ontology.

4 A GENERIC ONTOLOGY TO MASS CUSTOMIZATION

A Generic Enterprise Model (GEM) is a library of predefined concepts that are generic to the particular type of enterprise (Fox/Gruninger, 1998). This article elaborates on the description of a GEM that emphasizes on the specifics of MC in general and the arising cooperation issues in particular. Therefore a semi-formal, frame-based ontology (the *MC Ontology*) is proposed, providing terminology, taxonomy as well as some basic axioms significant to MC. This framework is intended to enable support for the following objectives:

- offer a description based on a suitable formal conceptualization, to provide a generic model, that is extensible, comprehensible and applicable,
- describe a domain ontology, that sufficiently considers specifics related to the "domain" Mass Customization to ease information interchange and reduce the coordination effort based on a common vocabulary,
- managing the complexity and consider strategic as well as operational aspects of the problem domain.

Furthermore, the model is meant to offer support to different groups of users, like software developers, domain experts and business consultants. Hence extension and refinement in ascertained projects is purposed to (1) achieve consensus among possible involved partners regarding the meaning of terms, (2) to enable knowledge exchange between all members along the supply chain network and finally (3) to simplify the implementation process for application developers. The latter one is realized by support of generic concept patterns that enable interoperability between information systems.
The next sections are used to discuss the efforts in constructing a Generic Enterprise Model based on a domain ontology that deals with aspects specific to MC. Furthermore it will be sown, how to create your own application ontology on top of this, by extending the predefined concepts, exemplary shown in the scope of the shoe industry up to the application level for a concrete mass customizing company.

4.1 Domain Ontology

The most general aspects of the MC domain are addressed within the *Generic Domain Ontology*. This top level layer subsumes concepts and relations that are valid for nearly all conceivable MC applications. Therefore the outlined ontology considers operational as well as the organizational aspects of a typical MC setting. This comprises the formal description of the main activities to cover the dynamic parts together with a representation of the (static) business objects to be modified. The specification is completed by gathering possible roles and institutions that are typically involved in the process of execution.

4.1.1 Extending the Enterprise Ontology

For creating an enterprise model it is necessary to describe basic concepts like *Activity*, *Plan*, *Product*, *Organization* or even common facilities like *Time* or *Event* first. Based on this, truly MC-specific elements like customer preferences have to be considered as well. First mentioned ones are already sufficiently described by existing so called *Top-level Ontologies* like the *Enterprise Ontology* (Uschold et al., 1997). This ontology-driven approach covers main aspects of an enterprise and enables support for further analyzes. The *Enterprise Ontology* therefore provides a glossary of terms and concepts significant to the enterprise domain. To capture the various aspects of an enterprise, the approach discerns five major categories: *Meta-Level*, *Activities* and *Processes*, *Organization*, *Strategy* and finally *Marketing*. Terms closely related to each other are attached to one category. Actually, the vocabulary used in the context of MC was created by specialization of particular terms introduced in this high-level ontology, since the defined concepts just belong to our universe of discourse as well. This made it possible to concentrate on the aspects that typically belong to MC. Consequently the following remarks entirely focus on the description of some selected MC-specific categories. Basic concepts are only referred if necessary.

4.1.2 Components of the MC ontology

For each element of the MC Ontology introduced within this article a short natural language description is given. Furthermore, to provide a better understanding, we have added some illustrations. It is our intent to present the reader a coarse overview about the main aspects concerning an enterprise model to MC, although it may seems to be rather informal. However, a more precise frame-based definition of the taxonomy is supported, too. This formalization is based on the *Meta-Ontology* described in (Uschold et al., 1997), a part of the *Enterprise Ontology* that itself is expressed in *Ontolingua*[1] a *KIF* derivation.

A note on formalization and terminology
For better readability and clearance, concepts (objects) start with a leading upper-case letter followed by lower cases, whereas slots (representing some binary relation) as well as functions are denoted in small letters only. Beyond all elements of the ontology are presented in *italics*.

Strategies
Products can be customized in different ways. We can distinguish five fundamental methods for a mass customizing company to select (Pine, 1993):

- Producing standard products that customer can easily adapt to their individual needs.
- Creating new customizable products around existing standard products.
- Moving the production along the value chain to the customer to provide point-of-delivery customization.
- Modularization of components to customize final products.
- A quick and synchronized response throughout the value-added-network.

For a concrete MC-strategy you need to utilize one or more of these methods in the right mixture. Because these methods are significant to MC we need to capture them as part of the ontology. Within the enterprise model, a *Strategy* is defined as a *Plan*, whose *Intended-Purpose* is a *Strategic-Purpose* (Uschold et al., 1997). Accordingly these methods where represented by means of *Customization-Strategies*, because that is what they are actually. Beyond this, objectives, processes, activities as well as the structure of the whole value chain needs to be adjusted and therefore has to consider the specifics of MC. The next paragraphs are used to discuss some of these specifics in detail.

Purpose
Everybody needs a motivation to take some action. Such intentions are be represented by the concept of *Purpose* (see Figure 3). Within the scope of the *Enterprise Ontology* this category is refined into further sub-concepts like *Vision*, *Mission*, *Goal* and *Objective* as well as *Critical-Success-Factor* and *Strategic-Purpose*. In the domain of MC we have some *Critical-Success-Factors* that seem to be especially crucial to the success of a MC company such as to *Meet-Customer-Demand* and making *Effective-Use-of-Resources*. On the other hand there are universally valid *objectives* like the *Reduction-in Lead-Time* or to *Minimize-Inventory-Costs*, both examples for a purpose with a defined measure. We had to consider such objects within our enterprise model to lay foundation for derivation of concrete strategies, plans and actions.

---

[1] The complete specification of the *Enterprise Ontology* as well as a detailed introduction to *Ontolingua* is available at the *Knowledge Systems Lab (KSL)* under http://www-ksl.standford.edu/.

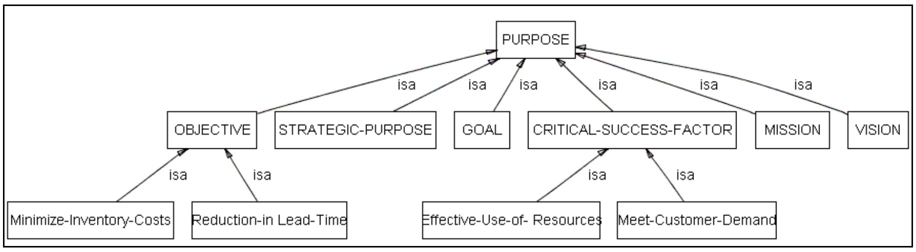

Figure 3: Purpose Hierarchy (detail drawing)

Plans and Processes

A plan can be understood as a specification of one or more *Activities*, a characterization of something to do (Uschold et al., 1997). A *Plan* is defined as an abstract concept that restricts range of possible *Activities* in the universe. Beside its constraint function a *Plan* can be decomposed into *Sub-Plans*. The term process is not explicitly used within the conceptualization of the *Enterprise Ontology* because of the amount of (possible) misleading interpretations this word is associated with normally. Instead a process specification (*Process-Spec*) is intended as a specialization of a *Plan* that is intended or at least capable of being executed more then once (Figure 4).

As MC clearly focuses on the customer the order, fulfillment raises special attention within a framework to this domain. A common fulfillment process to MC consists at least of a *Configuration* step to find a individualized product-specification together with the customer, followed up by the *Order-Preparation*, which contains activities like the arrangement of incoming purchase orders and forwarding of customer information to supply-chain members, the procurement or *Sourcing* of raw materials and accessories required, the transformation of the materials and components into an end-product within the *Production*, warehousing and hand over of the end-product commonly known as *Dispatching* and finally the *Transportation* process itself that covers logistics activities by means of shipping the end-product to the customer. These processes are completed by supplementary activities like the *Inventory-Management*, as the process of mapping demand forecast into stock of inventory and further after sales services for establishing a long termed partnership with the customer by Customer Relationship Management *CRM.*

Of course, most of these processes are not exclusively restricted to MC. Anyway, they are necessary. Within MC, nearly each activity has to regard the specific customer preferences and therefore is customer centered in some way.

Who takes care for concrete process is thereby expressed within the *execute*-relation, which binds a particular *Actor* to a *Process-Spec*. Thus, management of inventory for example can be fulfilled by a manufacturer, a logistics provider or even a supplier (like *KANBAN* approach - an integrated *Just-In-Time* concept for automated manufacturing).

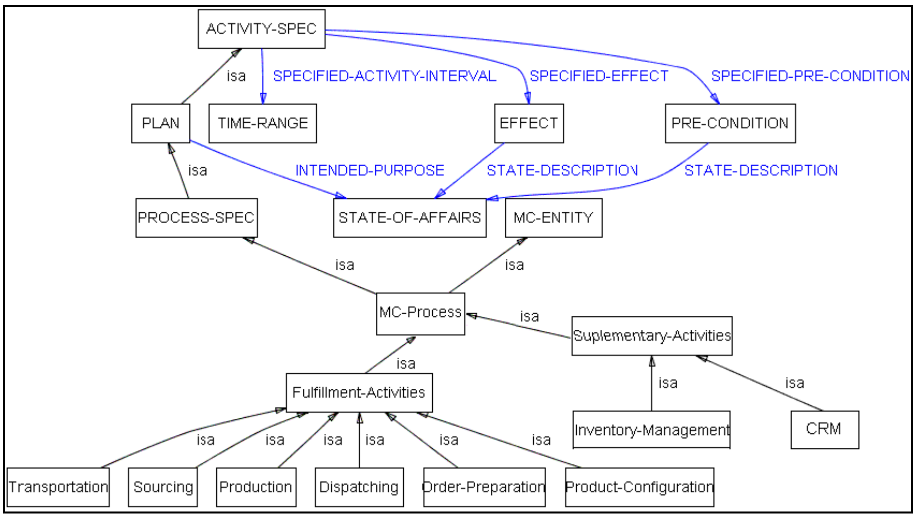

Figure 4: Taxonomy of common MC processes (detail drawing)

Activities
An *Activity* is defined as something that is done over a particular *Time-Range*. It could be decomposed into detailed *Sub-Activities*, get performed by one or more *Actors* – explicitly declared within the *Doer(s)* role. Furthermore, a particular authority is discerned to be responsible for an *Activity*. Formally this can be expressed inside of the *activity-owner* relation (Uschold et al., 1997). Resources are attached to an activity as part of the *can-use-resource* relation. There are mainly customer-faced activities like *Product-Configuration* which takes place within the sales process or *Order-Tracking* that is intended to get the customer up to date about the degree of order completion. Beside this MC, can be understood as a global process that requires the implementation of effective fulfillment processes, in order to reduce costs and lead times. Because of this, we need to recognize activities like *Stock-Level-Management* along the whole supply chain network, *Order-Processing* and *Global-Capacity-Planning* as well. Furthermore, the *Assembly-of-Components* has to be considered explicitly, due to the high modularization level within MC-fabrication.

Product
Since we have discussed dynamic elements particular to MC, we next want to light up some aspects concerning the business objects to be modified. First of all each *Product* is endowed with one or more *Features*. A *Feature* is defined as an attribute of a product. It possibly can be used to satisfy a customer need that depends on a particular functionality or a specific util.
Beyond this, there are some significant drawbacks regarding the structure and properties of products and services within MC domain. A mass customizing company must be able to support as many product variations as possible. Consequently the degree, to which a product is composed from modular component parts, is one important property of customized products. This property is called *Divisibility* (Benaroch, 1996). Furthermore such a product naturally has a lot of attributes. Hence, each product has some specific degree of customizable attributes in comparison to other products. This property is denoted by the term *Inter-Variability* (Benaroch, 1996). The *Intra-Variability* finally can be used to say something about the degree, to which the attributes themselves can be customized.
Profiles of individual customer's products need to be available as well. Such a *Product-Profile* offers in addition to profiles of the individual customer's important information about the customer, especially useful for Customer-Relationship-Management, to enable cross- and up-selling.

Components and Modularization-Schemes
Closely related to *Products* are *Components*. A *Component* is some prefabricated entity, as possible part of a product, assembled within the production process. These entities are used within the *consist-of* relation to express that some *Product* is created from a *Set-of-Components*. To define a set like this, we have to create an instance of the meta-class *Set, whose* members are restricted to be *Components* only[2]:

$$\forall Cs :(\textit{Set-of-Components}\ (Cs) \leftrightarrow \ \textit{Set}\ (Cs) \wedge \ \forall x: \textit{member}(x, Cs) \rightarrow \textit{Instance\_of}(x, \textit{Component}))$$

Now we can define an according *consist-of* relation:

$$\forall P, \exists S \in M :(\textit{Product}\ (P) \wedge \textit{Set-of-Components}(S) \wedge \textit{consist-of}\ (P, S)),\ |M| = 1$$

You can build own products by using components in different ways. This leads to slightly different production processes. In (Pine, 1993) according modularization schemes have been introduced therefore, such as *Component-Sharing*, *Component-Swapping*, *Cut-to-Fit*, *Mix*, *Bus* and *Sectional*. Because of their importance to MC, we decided to consider these concepts within our model.

Customer-Profile
Within MC it is necessary to treat each customer as well as each product on an individual basis. Therefore we need to have detailed information about the customer and its preferences along the whole value chain. To meet such a requirement the ontology offers an according concept – the *Customer-Profile*. This entity is used to share common information about a customer, like its shipping address or payment information as well as some more specific information, i.e. possible affectations regarding purchase facilities, configuration patterns etc.

---

[2] This formalization is proposed in (Uschold et al., 1997). For all remaining *Set*-entities an according definition is assumed implicitly.

Transfer Objects & Flows

A *Flow* is defined as an entity, which transfers an object (intangible or material) without modifying it between two successive activities. An *Activity* serves as start-point as well as end-point to *Flow*-objects. Hence every *Flow* starts with a particular *Activity* and ends in another *Activity*:

$$\forall f, \exists a_1, \exists a_2: (Flow\ (f) \wedge Activity\ (a_1) \wedge Activity\ (a_2) \wedge (start\ (f, a_1)) \wedge (end\ (f, a_2)) \wedge a_1 \not\subset a_2)$$

Similar to an *Activity,* which is described by means of an *Activity-Spec,* each *Flow* is described by an according *Flow-Spec,* which characterizes a flow. Thus it restricts the possible objects that can be transferred with this *Flow* as well as the way objects are handled by the *Flow*. We can explicitly discern between *Information-Flow* and *Monetary-Flow* as intangible and the physical *Material-Flow as* material *Flow-Spec* objects.

To support information interchange along the supply chain various *Information-Objects* are necessary like *Purchase-Order*, *Production-Order, Delivery-Note* and *Technical-Document*. As part of the *Material-Flow* we have to consider entities to be used within the production process as input factors *(Resources)* respectively output-factors *(Products).* Latter object type could be a *Good*, *Service* or even a monetary value (*Money*). Information, monetary and physical objects are subsumed under the abstract concept *Transfer-Object*, an entity that is forwarded across a spatial and temporal gap between two different *Potential-Actors.* Furthermore we distinguish between a *sender* and a *receiver* on a *Transfer-Object* as possible role types.

Interfaces

When describing the sub systems and their relationships of a supply web to MC, essential elements we have to consider are the possible interfaces that have to be provided. Only with a clear knowledge about interfaces that are available, we can ensure reliable and sufficient coordination.

There are to basic types of interfaces we have to discern – *Hardware-Interfaces* and Software-*Interfaces*. A *Software-Interface* gives you access to an ERP-system or CRM-system for example. Possible Hardware-Interfaces on the other hand could be a loading plant or a pick up station (where the Customer has to come for the good he had ordered). Hence, the question which type of interface to be used in a particular interaction mainly depends on the object that should be transferred. Again, we have to distinguish between a concrete interface and its specification. The specification to an *Interface* is offered by an *Interface-Spec.* To express that a certain type of object can be handled by a particular interface a new relation type was introduced. So, there are one or possibly more objects to be supported for an *Interface* that can be described with:

$$\forall I, \exists o: (Interface\text{-}Spec\ (I) \wedge Transfer\text{-}Object\ (o) \wedge (supports\ (I, o)))$$

Furthermore objects <u>can</u> be directed, means that is possible to use some element either as *Input-Paramete*r or as *Output-Parameter*. Finally we have to distinguish between whom is the caller of Interface despite from the fact who owns it:

$$\forall I, \exists c, \exists o: (Potential\text{-}Actor(c) \wedge Potential\text{-}Actor(o) \wedge Interface\ (I) \wedge caller\ (I, c) \wedge owner\ (I, o)))$$

It is possible of course that caller and owner of an interface are the same (as for an internal information system). Therefore we decided not to restrict these roles by means to be distinct.

Decoupling Point

Within MC different levels of customization take place on the continuum of mass production to pure customization. According to the level of customization we have different *Modularization-Strategies, e.g.*:

- *Deliver-to-Stock*. Customer is not involved. The product is picked-up from the shelf. This strategy is typical to mass production of standard products.
- *Deliver-to-Order*. The product is delivered to when ordered from the stock.
- *Assemble-to-Order*. Assembly of components and raw material does not start without a concrete customer order.
- *Fabricate-to-Order*. Only raw materials are stored to the stock. Fabrication is done to order.
- *Design-to-Order*. Customer is already involved at the design stage. No inventory is held.

Depending on the level of customer involvement within fulfillment process there are customer-specific steps as well as activities that not depend on a particular customer. The point of customer involvement within the manufacturing process is called the *Decoupling-Point.* Each *Activity-Spec* can either be attached to the *Post-Decoupling-Point-Activity-Set* or the *Pre-Decoupling-Point-Activity-Set.*

Actors

MC depends of several independent *Actors.* Some of them are *Legal-Entities* like *Vendor*, *Retailer*, and of course the *Customer*. Furthermore, there are other entities like *Machine* or *Organizational-Unit* for example, which can also be *Potential-Actors* to many roles within an Enterprise. Together these entities attend in the execution process of MC.

Some actors are already defined as part of the *Enterprise Ontology*. Other actors had been added like the *Forwarder,* who is responsible for distribution of the end-product or the *Supplier,* that supports *Raw-Materials* respectively *Components.* Often the product vendor is not same as the producer of the product. Because of these, the *Manufacturer* concept had been introduced to the ontology. Another important *Potential-Actor* for the MC domain is the *Configurator.* Because this facility enables the customization process it has to be considered as well. A concrete configuration system and therefore a *Configurator* can be shaped very differently. This mainly depends on the product or service that should be customized. Configuration can take place by a shop assistant that measures your feed or in terms of a web side, for example.

Supply-Chain-Network

As already mentioned, within MC, there is a high demand for efficient coordination of actors along the value chain. This is caused by the natural distribution and heterogeneity inherent to this domain. Transitions between the particular *Activities*, i.e. between *Actors* associated to those *Activities*, are sufficiently described by different types of *Flow*-objects. Information as well as material interchange is gathered by means of the *Interface* concept, whereby *Transfer-Objects* are used to characterize the objects of interest. To be missing is a further conceptualization concerning potential relation types and their possible temporary implications.

The concepts introduced within the Enterprise Ontology mainly focuses on a single enterprise. But, as we need to describe groups of interacting companies in the context of MC, where, we had to define some topology related concept first. Therefore, we added the *Supply-Chain-Network (SCN)* concept, as a temporary confined structure of *Legal-Entities,* which share a common *Goal.* Consequently, such a network consists of a bunch of *Partners*. Moreover, we need to define some kind of *Contract* that bindingly describes the duties and rights of each of these *Partners*. Each natural and artificial person can fill this partner role.

To represent an enterprise the *Corporation* concept is used. A *Corporation* is defined to be a group of *Persons* who "are recognized in law as having existence, rights, and duties distinct from those of the individual Persons" (Uschold et al., 1997). Furthermore, by means of the *Partnership* concept we can state that there is a group of *Persons* that carrying on the same thing or task. While *Partners* if you think of a *Customer* or even a *Supplier* for example may come and go and a *Partnership* consequently represents a temporary state, the *Supply-Chain-Network* itself is continuing over time (but with possibly changing members). However, no statement is made currently about the duration or conditions for start and end of a *Partnership.* To support such information we need to define further objects, but this is not subject to our ontology so far.

Performance Attributes & Metrics

Within MC, activities are linked together dynamically according to the product and customer profiles. There is normally one particular item, which has to be delivered corporately by retailer, vendor, manufacturer and suppliers within an acceptable time frame - acceptable to the final customer. Hence, it is essential for each vendor to specify performance criteria that (a) provide basis for contracts and (b) enable controlling of all value chain activities and members.

In 1996 the *Supply Chain Council* has developed and published a cross-industry standard for the management of supply chains, the *Supply Chain Operations Reference Model (SCOR)* (Stephens, 2000). To measure respectively control the efficiency of a value chain several performance attributes are proposed, to prove the reliability, responsiveness, flexibility and costs as well as the asset management of a supply chain. For means of quantification, each attribute is linked to some specific metrics as shown in table 1 (Stephens, 2000):

Performance-Attribute	*Metric*
SC-Delivery-Reliability	*Delivery-Performance, Fill-Rates, Perfect-Order-Fulfillment*
SC-Responsiveness	*Order Fulfillment Lead Times*
SC-Flexibility	*SC-Response-Time, Production-Flexibility*
SC-Costs	*Cost-of-Goods-Sold, Total-Supply-Chain-Management-Costs, Value-Added-Productivity, Warranty/Returns-Processing-Costs*
SC-Asset-Management-Efficiency	*Cash-to-Cash-Cycle-Time, Inventory-Days-of-Supply, Asset-Turns*

Table 1: Performance attributes and their metrics

To enable controlling and benchmarking on a supply web for MC, we have adopted the illustrated performance attributes together with their associated metrics within the framework.

4.2 Using the MC Ontology

One task of the University Hohenheim within EwoMacs is to develop a tool to model the supply chain and develop tools for agent-based simulation. In our prototypic approach we implemented have already implemented parts of the described MC ontology. Actors and products have been in the focus. In detail, the following actors have been realized: customer, vendor, producer, supplier and carrier. In contrast to the developed MC ontology the forwarder and manufacturer has been renamed to carrier resp. producer. With respect to simplification and functionality we decided to integrate the configurator into the customer’s role. This is one result of high customer’s orientation and integration during the configuration process in the context of MC. Since we talk about the shoe industry, there is a need to specify shoes in particular. In order to apply the product and related component concept, we introduce a closer look to our shoe description. A shoe is a *product* and therefore it exist a *Set-Of-Components* with the following specific *components*: leather, sole and smallparts. *Features* of a shoe or in general a pair of shoes are color, leathertype and soletype. Based on these information agents can deal with shoes or even the components of shoes. In Figure 5 the entire supply-chain is illustrated and here the *Decoupling-Point* is assigned to the *producer*.

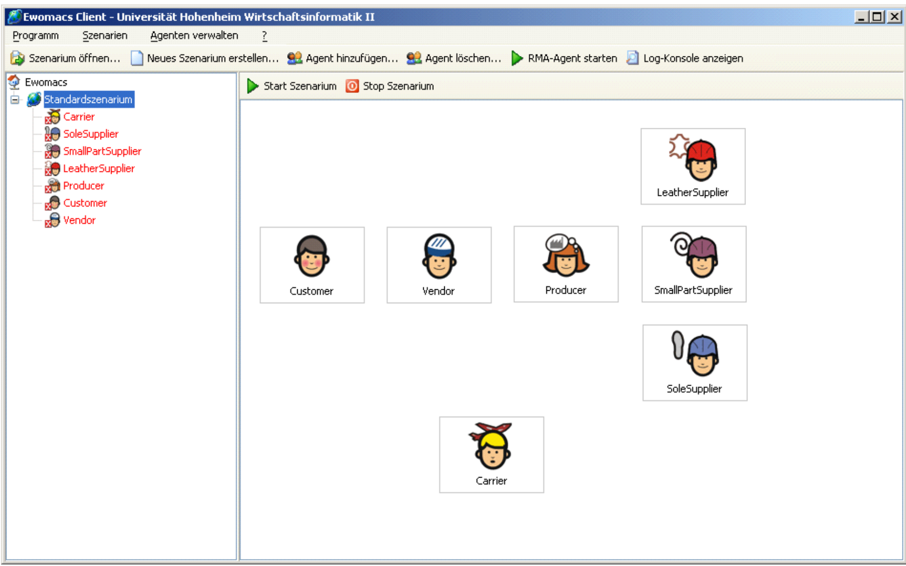

Figure 5: EwoMacs-Client

As it has been pointed out, MC impacts on enterprise strategies, product design, manufacturing and assembly processes as well as on logistics and information processing. Consistently we have presented a generic domain ontology that deals with these specifics in general. For evaluation purpose the generic domain ontology had been applied to a particular domain. In this section we want to discuss the experiences we have gained, while testing our framework.

The EwoMacs Project

Evaluation of the ontology took place as part of the joint research project EwoMacs (Dietrich et al., 2003b). Within this project, logistics structures of MC in the shoe industry are analyzed. One objective of this project is to deliver a business model for an optimal logistics structure of MC. Therefore relevant parts of the supply chain are simulated. As already mentioned, we can think of a supply chain as a group of legally and economically independent companies represented as nodes in a network, that exchange material and data flows to fulfill a common goal. Simulation is used in this context to observe selected economical magnitudes like as delivery time, unit cost prices, and reliability. From the simulation results one can make an appraisal about the overall system performance of the business model. On the other hand validation and verification of existing business processes can take place, by testing alternative supply chain strategies. Beside the simulation part the projects aims to provide common knowledge based co-operation procedures for controlling the information flow on supply chain level. The integration of a MC-Ontology should guarantee that the software target concept can be adapted to different MC applications.

MC within the Shoe Industry

Nowadays, shoe industry is still characterized by little degree of automation and lack of modern production and logistics processes. In the context of the EwoMacs project, two companies with different starting points will be analyzed. Selve AG (Germany) has been founded for implementing a business model for mass customized ladies shoes. All internal processes are especially designed for handling the MC specific process steps. In contrast to that, adidas-Salomon AG (Germany) is serving as a representative enterprise with its MC-project “mi Adidas”. Here, MC has to be integrated into well-established mass production processes.

Implementation

To manage the job of simulating the supply chains of Adidas-Salomon AG and Selve AG, we had to capture the typical system behavior of a MC system for the shoe industry. Thus, there was a need for a description technique to envelope the dynamic aspects of applied business processes and structure. To sketch the problem domain by analytical means we decided to use an ontology driven approach. Therefore we have adapted the developed MC-Ontology by adding further concepts, specific to the domain of shoe industry. Within this conceptual model all entities and relations are described and solidified at an abstract level. In the resulting ontology business goals and policies of the system as well as the organizational environment were covered sufficiently. The resulting model relates to the structure within the application domain and covers issues such as the type of persons, activities, interfaces, flow of information and goods in the respective domain. The model also contains some normative component of the system such as permissions, rights, and norms.

Beyond this, the ontology is part of the application system itself. To simulate the typical behavior of logistics processes for example, we need to consider situations like negotiation or decision making. For such situations we have to provide detailed knowledge about the domain, to support the same decisions and to have the same behavior as the real world system to be represented. Such pre-requisite knowledge is held inside of an ontology as well.

The ontology modeling was done in Protégé, a tool widely used for ontology design, that offers graphical representation capabilities. Our description model therefore is intuitively understandable and grounded on a semiformal semantics. The notation is equally adequate for the demands of application domain experts as well as system engineers.

5. RELATED WORK

We explicitly argued for the need of a common semantic foundation to enable efficient interoperation between distributed information systems. Up to now, there are different projects that deal with the use ontologies for information systems. Some collaborate in mediation services like information broker architectures that aim on accessing and retrieving information from multiple distributed resources

(Fikes/Farquhar, 1996). In the recent past, information and service retrieval is discussed in connection with Semantic Web approach (Vögele, 2003). Others endeavor in development of ontologies to support communication of intelligent software agents, whereas agents have to commit on common ontologies and terminologies (Chandra, 2000). Moreover there are approaches that provide generic ontologies for knowledge management in enterprises, to offer a formal representation that can be shared among different software systems (Fox, 1993; Gruninger/Fox, 1996; Fox, 1997). In (Guarino, 1998) an "Ontology driven Information System" is discussed, where the ontology lays foundation to all aspects and components of an information system.
Within this article we pointed out on a generic domain model to MC. Therefore we have extended the *Enterprise Ontology*, a generic enterprise model (Uschold et al., 1997). There are several contributions related to enterprise models, like the *Enterprise Ontology* in common. A good overview on existing formalization approaches to enterprise models can be found in (Fox/Gruninger, 1998).
Furthermore, there are some attempts to exploit particular aspects of MC by means of ontologies. In (Ardissono, 2001) for example an approach to customer-adaptive and distributed online product configuration is discussed. The reuse of knowledge in the context of knowledge-intensive services for MC is introduced in (Benaroch, 1996).

6. CONCLUSION AND FURTHER WORK

In this paper, we propose an ontology for Mass Customization in order to support this management concept in a formal manner. After a short introduction into the foundations of MC, semantic interoperability, and ontologies we introduced a generic MC ontology. Some of the main parts of this MC ontology are models for existing entities and their properties and relations: actors' interactions, product architecture, manufacturing structure, and role assignment. For building this model, results of process modeling efforts within the EwoMacs projects has been used. The MC ontology is intended to be applied within an agent-oriented simulation system, primary. Nevertheless, the ontology could be used to support real application. The ontology approach is not limited to agent-based systems. In the context of web services, it is possible to implement systems using semantic-based interoperability without explicit application of agent technology. In this paper, the common vocabulary and the interactions among the actors are described. Beneath that, the ontology should optimize the inter-organizational cooperation in general. Although MC was introduced as competitive strategy it causes fundamental changes in enterprises. This does not appear on a strategic level only, but on the operative units, too. So, research and development, marketing, production, distribution etc. within one enterprise and the relationships and interactions with other organizations must be reengineered. This opens a visionary aspect of the approach: The MC ontology should support a better understanding about MC in order to be successful in implementing MC business models.

The work on building a MC ontology has not finished yet. So, there are several open tasks. For this future work section, we want to point out three main aspects:

- **Formalization and implementation**: The MC ontology must be transferred in appropriate modeling languages (e.g. DAML, DAML+OIL, RDF, OML, OWL etc.) using tools like Protégé for example.

- **Domain specification**: As the shown approach is divided into two parts (generic MC ontology and domain specific MC ontology) several case studies can be used to adapt the generic MC ontology for the requirements of a particular domain. In this paper, a first example for shoe industry was given.

- **Evaluation**: Finally, the benefits of usage an ontology for MC scenarios must be evaluated. On the one hand technical systems that are using the ontologies must be tested. But one other hand the level of abstraction and the included components must be evaluated with domain experts in order to ensure that all relevant aspects of the particular domain has been taking into consideration.

ACKNOWLEDGEMENT

This work has been funded by the German Federal Ministry for Education and Research (BMBF) within the research project "EwoMacs" (02PD1121). Further information is available on the project website: http://www.ewomacs.de.

REFERENCES

Ardissono, L.; Felfernig, A., Jannach, D.; Friedrich, G.; Schafer, R.; Zanker, M. (2001): *Customer-Adaptive and Distributed Online Product Configuration in the CAWICOMS Project,* 17th *International* Joint Conference on Artificial Intelligence - Configuration Workshop, Seattle, USA, August 2001.

Benaroch, M. (1996): *Knowledge Reuse in Mass Customization of Knowledge-Intensive Services,* Artificial Intelligence in Economics and Management, Ein-Dor P. (Ed), Kluwer Academic Publishing, Boston, pp. 107-127.

Becker, M.; Heine, C.; Herrler, R.; Krempels, K.-H. (2003): *OntHoS - an Ontology for Hospital Scenarios,* In: Moreno, A.; Nealon, J. (Eds.): *Applications of Software Agent Technology in the Health Care Domain*, Whitestein Series in Software Agent Technologies (WSSAT), Birkhäuser Verlag, Basel pp. 87-103.

Brachman, R. J.; Schmolze, J. G. (1985): *An overview of the KL-ONE knowledge representation system*. Cognitive Science, 9(2), pp. 171-216.

Chandra, C.; Smirnov, A.; Sheremetov, L. (2000): *Agent-based Infrastructure of Supply Chain Network Management*. In: Camarinha-Matos, L.M., Aarmanesh, H. and Rabelo, R. (Eds.) *E-Business and Virtual Enterprises Managing Business-to-Business Cooperation,* Kluwer Academic Publishers, pp. 221-232.

Cranefield, S.; Purvis, M. (1999): *UML as an Ontology Modelling Language,* The Information Science Discussion Paper Series Number 99/01.

Dietrich, A. J.; Timm I. J.; Kirn, S. (2003a): *Implications of Mass Customization on Business Information Systems.* Proceedings of the Mass Customization and Personalization Congress 2003. Munich, October 2003.

Dietrich, A. J.; Pawlaszczyk, D.; Kirn, St. (2003b): *Agentenorientierte Simulation von logistischen Prozessen in der kundenindividuellen Massenproduktion*. In Inderfurth, K.; Schenk, M.; Wäscher, G.; Ziems, D. (Hrsg.): *Logistikplanung- und management. Begleitband zur 9. Magdeburger Logistik-Tagung „Logistik aus technische rund ökonomischer Sicht"*, 20. - 21. November 2003 Magdeburg. pp. 198-212.

Fox, M. S., Barbuceanu, M.; Gruninger, M.; Lin, J. (1997): *An Organization Ontology for Enterprise Modelling*.

Fikes, R.; Farquhar, A.; Pratt, W. (1996): *Information brokers: Gathering information from heterogeneous information sources*. Stanford University.

Fox, M.S.; Chionglo, J. F.; Fadel, F. G. (1993): *A Common-Sense Model Of The Enterprise*, In Proceedings of the 2nd Industrial Engineering Research Conference, volume 1, pages 425--429, Norcross GA, USA.

Fox, M. S.; Gruninger, M. (1998): *Enterprise Modelling.* The AI Magazine, Fall 1998, pp. 109-121.

Gilmore, J. H.; Pine II, B. J. (1997): The Four Faces of Mass Customization, Harvard Business Review, Vol. 75, No. 1, pp. 91-101.

Gruninger, M.; Fox, M.S. (1996), *The Logic of Enterprise Modelling, Modelling and Methodologies for Enterprise Integration*. Bernus, P.; Nemes, L. (Eds.), Cornwall, Great Britain: Chapman and Hall.

Gruber, T. R. (1992): *ONTOLINGUA: A Mechanism to Support Portable Ontologies*, technical report, Knowledge Systems Laboratory, Stanford University, Stanford, United States. see: http://citeseer.ist.psu.edu/gruber92ontolingua.html, access: 2004-02-01.

Gruber, T. R. (1993): *A translation approach to portable ontology specifications,* In: *Knowledge Acquisition*, An International Journal of Knowledge Acquisition for KnowledgeBased Systems, 5 (2). June 1993, pp. 199-220.

Guarino, N. (1998): *Formal ontology and information systems*. In Guarino, N. (ed.), *Formal Ontology in Information Systems*. Proceedings of the First International Conference (FOIS'98, June 6-8, Trento, Italy) Amsterdam 1998, pp. 3-15.

Holsapple, C. W.; Joshi, K. D. (2002): *A Collaborative Approach to Ontology Design,* Communications of the ACM. Vol. 45, No.2, February 2002.

Jones, D. M., Bench-Capon, T. J. M. and Visser, P. R. S. (1998): *Methodologies for Ontology Development,* In: Proceedings IT&KNOWS Conference, XV IFIP World Computer Congress, Budapest.

Knowledge Interchange Format (1998), draft proposed American National Standard (dpANS) NCITS.T2/98-004.

Kim, W.; Seo, J. (1991): *Classifying schematic and data heterogeneity in multidatabase systems*. IEEE Computer, 1991. 24(12), pp. 12-18.

Knirsch, P.; Timm, I. J. (1999): *Adaptive Multiagent Systems Applied on Temporal Logistics Network*. In: Muffatto, M.; Pawar, K.S. (Eds.): *Logistics in the Information Age*. In: Proceedings of the 4th International Symposium on Logistics (ISL-99), Florence, Italy. Padova 1999.

Mertens, P. (1995): *Mass Customization (Massen-Maßfertigung).* In: Wirtschaftsinformatik 37 (1995) 5, pp. 503-506.

Müller, H.-J. (1997): *Towards Agent Systems Engineering*. In: Data & Knowledge Engineering 23 (1997), Nr. 3, pp. 217-245.

Obitko, M.; Marík, V. (2002): *Ontologies for Multi-Agent Systems in Manufacturing Domain,* In: Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002). IEEE Computer Society, Aix-en-Provence, France 2002, pp. 597-602.

Ontoweb Ontology-Based Information (2002): IST Project IST-2000-29243: *OntoWeb - Ontology-based Information Exchange for Knowledge Management and Electronic Commerce D21 Successful Scenarios for Ontology-based Applications v1.0,* see: http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/ OntoWeb_Del_2-1.pdf; access: 2004-03.20.

Piller, F. T. (2003): Mass Customization - Ein wettbewerbsstrategisches Konzept im Informations-zeitalter. Wiesbaden 2003.

Pine, B.J. (1993): *Mass Customization*. Harvard Business School Press, Boston 1993.

Piller, F. T.; Stotko, Ch. M. (2002): *Mass Customization: four approaches to deliver customized products and services with mass production efficiency*. In: Proceedings to the 2002 IEEE International Engineering Management Conference. Managing Technology for the New Economy, 18 - 20 August 2002: St. John's College, Cambridge, UK. Cambridge, pp. 773-778.

Porter, M. E. (1999): *Competitive strategy: techniques for analyzing industries and competitors*. New York 1980.

Sugumaran, V.; Kirn, St.; Dietrich, A. J. (2003): *Towards an Agent-Based Mass Customization Environment: Architecture and Coordination*. In: Proceedings of the Second Workshop on e-Business WeB2003. Dezember 2003. Seattle, USA. pp. 329-335.

Smirnov, A. V.; Chandra, Ch. (2000): *Ontology-Based Knowledge Management for Cooperative Supply Chain Configuration.*

Stephens, S. (2000): *Supply Chain Council & Supply Chain Operations Reference (SCOR) Model Overview*, see: http://cob.isu.edu/mba691/SCORoverview.pdf, access: 2004-03-10.

Timm, I. J.; Herzog, O.; Tönshoff, H. K.; Woelk, P.-O. (2001): *Akzeptanz von Agententechnologie in der industriellen Anwendung - Fortschritt durch Transparenz und Standardisierung?* In: Industrie-Management, Ausgabe 6/2001 (Jg. 17), GITO-Verlag, Berlin 2001, pp.13-16

Uschold, M.; King, M.; Moralee, S.; Zorgios, Y. (1997): *The Enterprise Ontology*, see: http://www.aiai.ed.ac.uk/project/pub/documents/1998/98-ker-ent-ontology.ps, access: 2004-01-27.

Uschold, M. (1996): *Converting an Informal Ontology into Ontolingua: Some Experiences*. ECAI, Budapest 1996.

Uschold, M. (2001): *Barriers to e_ective agent communication*. In: OAS'01 Ontologies in Agent Systems [6].

Uschold, M; Gruninger, M. (1996): *Ontologies: principles, methods and applications*, Knowledge Engineering Review, Vol.11, number 2, pp. 93-155.

Vikram N. K; Whitman, L.; Malzahn, D. (2003): *Ontology-based Product Tracking System*.

Vögele, T; Huebner, D.; Schuster, G. (2003): *BUSTER - An Information Broker for the Semantic Web*. Künstliche Intelligenz Heft 3/03, arendtap Verlag, Bremen, pp. 31-34.

Wache, H. (2003): DISKI - Dissertationen zur Künstlichen Intelligenz. Bd. 261: *Semantische Interpretation für heterogene Informationsquellen*. Köln 2003.

Weiss, G. (Ed.) (1999): *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. Cambridge, Massachusetts 1999. see: http://citeseer.ist.psu.edu/80667.html, access: 2004-02-01.