A Framework for the Development of Multi-Agent Systems in Supply Chain Management

Bernd Hellingrath  
Fraunhofer Institute for Material Flow and Logistics  
University of Münster  
bernd.hellingrath@iml.fhg.de  
hellingrath@ercis.uni-muenster.de

Carsten Böhle  
University of Paderborn  
carsten.boehle@hni.upb.de

Jan van Hueth  
Capgemini Consulting  
jan.van.hueth@capgemini.com

Abstract

The application of multi-agent systems in the context of supply chain management has been much discussed during the last years and several implementations have emerged within research. Yet very few applications are actually being used in practice. The reason for this lies in an existing gap between domain concepts and agent technology, which complicates the application oriented development. A framework can support this process in order to devise customized and flexible solutions by providing the necessary agent modules and a supply chain management specific ontology. Within this paper, an analysis of requirements on multi-agent systems that stem from the domain of supply chain management is carried out. This forms the basis for the development of the architecture of the multi-agent framework whose design and implementation is described. In the following, common modules are designed and implemented to provide both a sophisticated and generic basis for specific projects.

1. Introduction

Multi-agent systems and supply chain management have both emerged during the last ten to fifteen years and have gotten a lot of attention from the computer science and management science research community, respectively. Researchers in supply chain management have long since favored multi-agent systems as a suitable modeling technique for the structures and processes they examine. It can be shown that multi-agent system features and supply chain management requirements match very closely, e.g. by employing the classification scheme of Russel and Norvig [1]. It uses the dimensions of observability, predictability, sustainability, reliability, continuity, and interaction density and spans a hexagon whose size is believed to correlate with the applicability of multi-agent systems. The measures are marked on each axis and rise in proportion to dynamics and complexity. If such a graph is drawn for the field of supply chain management, it can be seen that the resulting area of the hexagon is sufficiently large, although the actual size depends on subjective estimations. Still there is a tendency that markets alone are not fully observable, demands are stochastic, and that the economy is inherently decentralized and consists of a multitude of actors.

A good example are the automotive industry and its environment which are characterized by high dynamics and uncertainty caused by unsteady demand and new products. Root cause is the customer and consequently his demands on which the network has to react. A great deal of flexibility has to be maintained by all network members, especially the suppliers. Also the network has to frequently reassess paradigms and processes as well as to make changes to the structure when needed. Product life cycles are constantly shortening so that adaptivity becomes vital. A concrete example is the role of the logistics service providers which has evolved a lot in the last years. Having been a mere transporter in the past, these companies now take over an increasing share of responsibility, e.g. for packaging or quality tests. Further changes like the re-assignment of existing tasks or the emergence of new ones within a network that alters at an increasing pace have to be expected. The automotive industry will serve as a test case in the following.

There are many definitions of supply chain management, but they all roughly conclude that it is “an approach where the entire network from the supplier to the ultimate customer is analyzed and managed in order to achieve the best outcome for the whole system” [2]. In most cases, planning methods try to come up to this best outcome by means of
centralized planning. Although this is applicable in theory, there are numerous obstacles in practice. Networks consist of legally independent companies that are only loosely coupled regarding planning processes. There are severe reasons why an integrated centralized planning is not possible. Most companies are involved in more than one supply chain so that conflicts arise, e.g. concerning resource allocation. Also, central models require all data, including confidential information that companies are not willing to share as they fear fraudulent use. Additionally, giving up planning authority and accepting central plans might realize the best performance of the network, but it will most probably also worsen individual members’ margins. Last, organizing and accomplishing central plans is very hard to organize. It becomes apparent that planning methods that work well within a company are not suitable for inter-company planning. Here, methods have to be devised that take the problems mentioned into account. In particular, the method has to maintain planning authority for all members, has to limit information exchange to non-critical data, and has to formulate principles of plan alignment. As there will be different implementations in different companies, it is important to keep the formal concepts independent of their technical realization.

Multi-agent systems are a suitable basis for these methods. Several implementations have been designed with this intention [3] (cf. also sec. 2). However, they all share a common downside, i.e. they all have been devised for a specific purpose. In contrast there are generic agent platforms which do not assist domain specific modeling. Thus, specific features such as the bill of materials (BOM) or supplier relations have to be modeled again from scratch for each project. In summary, there are either platforms with no assistance regarding the application domain or inflexible implementations that cannot easily be adopted to particular needs. Having identified this lack, a framework that supports the flexible development of multi-agent systems in the domain of supply chain management will be presented in the following.

2. Multi-Agent Systems in Supply Chain Management

Several multi-agent systems that tackle supply chain management problems have been presented in the past. We have chosen three systems for this chapter. MASCOT represents one of the earliest approaches and is frequently cited, CoagenS is a system that is in practical use, and AgentEnterprise is a current and extensive project. There are more systems, but those are either scarcely documented or focus on a particular problem instead of covering a wide range of supply chain activities.

MASCOT (Multi-Agent Supply Chain Coordination Tool) [4] is described by the authors as “a reconfigurable, multilevel, agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling”. Multilevel means that MASCOT distinguishes between agents on different ranks in an organization’s hierarchy. Communication is possible for each agent vertically and horizontally. Vertical communication serves the purpose of aligning plans with other agents of the same or another company. Horizontal communication ensures consistency. Problem solving is based on the blackboard concept, which is praised by the authors as an approach that “explicitly separates domain knowledge […] and control knowledge”. Each agent possesses a set of so-called Knowledge Sources. These can be planning systems or data sources like ERP systems. Another interesting design aspect is the ability to assign certain decisions to the user. Apart from that, the user can always interfere and alter data and decisions. Unfortunately, apart from the general description of the architecture, no details on the implementation are available. MASCOT’s strengths are its connectivity to legacy planning systems and the possibility of user integration. Its weaknesses are the missing distinction of intra- and inter-company processes and most of all the lack of a description of abstract models and of how semantics are supported.

CoagenS (Cooperative/competitive agents for the organization and operation of production networks in Serial manufacturing) was developed at the Fraunhofer ALB. It aims particularly at improving the planning process from the logistics service provider’s point of view. For this purpose, an XML-based communication format was developed which enables messaging between buyer, carrier, and supplier. The new process determines the pick-up and delivery dates with regard to the carrier’s requirements. Interesting about this approach is the separation of the formal process description and its technical implementation. A process is expressed in an XML file. This file contains tags which match message IDs. Substructures give information on the Java function to be executed, the parameters, and the next process step. The system is built on top of Lotus Notes/Domino and communicates via e-mail. Mails are automatically handled if a CoagenS instance is running. Otherwise, a manual reply is possible. CoagenS has been tested in practice and delivered good results in terms of process integration and the reduction of manual work as well as the enhancement of the availability of data. Furthermore, this project made the first step towards
providing the possibility of modeling transparent business processes for domain experts without profound IT skills. This is factor that influences the solution’s acceptance and trust. The pivotal disadvantage here is the missing semantical foundation.

Agent.Enterprise [5] is the result of a project founded by the DFG (German Research Foundation) and is one of the newest systems. Technically speaking it is a Multi-Multi-Agent-System that distinguishes between intra- and inter-enterprise planning. The coordinating system is called DISPOWEB. It connects the different intra-enterprise planning systems and negotiations about orders, prices, and dates are conducted via its agents. The detailed planning on the company level is done by a single specialized system. Three different ones are available (IntaPS, KRASH, FABMAS) which focus on discrete manufacturing, assembly, and batch production. Supply Chain Event Management is also integrated (A4). In order to separate functional knowledge from network knowledge, the concept of gateway agents is introduced. Gateway agents help integrating complex systems because they set up a top-level multi-agent system based on a common ontology that specifies conversation semantics. An ontology is the formal specification of domain concepts and their relations. The individual implementations of functionality can use their own concepts as long as the gateway agent can translate messages between both worlds. The result is a decoupling of dedicated systems from the network system. It has been shown in a reference scenario that Agent.Enterprise in this case and multi-agent systems in general can help to cope with the inherent complexity of the application area and that they are able to increase flexibility. However, Agent.Enterprise does only provide assistance at the network level. There is no aid, e.g. in form of ontologies, for implementing new multi-agent systems below the gateway agents.

These three examples are exemplary for the state of the art. Recent advances have acknowledged the need for the formalization of domain concepts. A hindrance is the multitude of opinions about these very concepts. As commonly agreed upon definitions are not to be expected soon, systems aim at allowing local definitions that are mapped on a standardized conceptualization used for communication. Despite the advances made, topics can be identified that demand an in-depth analysis and discussion, foremost the question of how an ontology of supply chain concepts should look like. More on this topic in general can be found in Chaib-draa and Müller [6].

3. Cooperative Planning and Coordination Mechanisms

The need for cooperative planning was illustrated in Section 1. Different projects have dealt with means of coordination for processes and planning. Relevant developments will be presented in the following along with an analysis of how these can contribute to the proposed framework. A current project in the same field funded by the European Union is AC/DC. As this is ongoing work, results are not included here.

Odette started the project DCP (Demand Capacity Planning) [7] as part of its supply chain management endeavors. Their goal is to facilitate the acceptance of supply chain management concepts in the automotive industry. DCP is an important part of that assignment as experience shows that a lot of problems in the 1990s arose from badly aligned capacities. So the task is to formulate a common understanding of the subject matter and to define processes for a closer collaboration. The assumption is that capacity shortfalls and eventual sales losses can be avoided if capacity needs and shortages are communicated in time. The time horizon in this case is medium to long term and covers production planning and sales management rather than short term operations. For these periods it is checked whether demand and capacity match. If this is not the case, measures are taken at the supplier first. If those are not sufficient, a collaborative action plan is instantiated. The interesting aspect for this work is the matching of demand structures to resource structures. It is encouraged that demand is aggregated on different levels. This distinction is necessary to ensure an appropriate level of detail over the time concerned. The lowest level describes parts in detail over a time interval of one month. The same approach is used for resources. Allocation of items to resources is established on the lowest respective level and employs a coefficient for resource usage. This is called the Item-Resource-Link. The DCP level has to be transformed and mapped to the factory calendars and is an agreement to exchange time dependent data. Odette DCP also includes structures to describe adjustment measures including information about affected resources, effects, prerequisites, etc. The concepts on formalization of the BOM, capacities, their mapping, the time model, and the description of the properties of adjustment measures deliver valuable input to the framework.

LiNet [8] was a joint project of the German automotive industry aimed at mid and long term capacity planning in multi-level supply chains and short term delivery and material flow planning.
Besides other topics, the results on collaborative capacity and demand planning are of special interest for this work. The reason is the system architecture and its abstraction from actual implementations. The challenge is to enable decentralized processes for collaboration that are based on different ERP systems. For this purpose, LiNet defines a data model as an interface. Each instance has to map its data on this model to be able to communicate with other instances. The creation of a common understanding decouples ERP systems, optimization systems, and communication systems. However, except automatic forwarding of demands, LiNet relies on user interaction to solve capacity shortages. The idea of extending existing systems rather than replacing them and the separation of local and system-wide data models are technical guidelines for the framework.

ILIPT (Intelligent Logistics for Innovative Product Technologies) [9] is a European project that is active in many fields of research for the 5-day car. The main assumption is that production has to switch to built-to-order processes almost entirely throughout the network. One sub-project deals with flexible supply networks. It defines supply chain structures as being a combination of built-to-stock and built-to-order suppliers with only one order decoupling point allowed in the chain. Additionally, it introduces the Virtual Order Bank which is the central element in communication. It aggregates and surveys planning data and is an independent entity. The different actors in the supply chain consist of separate models that interact via a communication model. ILIPT contributes the neutral coordination entity to the framework’s specification.

The concepts of all these projects are based on automotive supply chains and the fact that every time several OEMs and suppliers participated ensures that they are sufficiently generic so that they can be referenced in the framework. Their input will reflect in the design of the architecture of both the agents and the network.

4. Framework Requirements

The framework must provide models and ontologies for the domain of supply chain management. They have to be designed in a way that allows extending and exchanging them. It has to accelerate and ease development significantly; otherwise it is no more useful than a generic platform. We will present functional and non-functional requirements in the following which provided a basis for the work on design and implementation.

4.1 Functional Requirements

The functional requirements derive from the properties of (automotive) supply chains as described in Section 1, the insights won from analyzing pros and cons of existing approaches in Section 2, and the state of the art in cooperative planning outlined in Section 3. They encompass those on: organization, data mapping, support of communication processes, and execution of processes for planning, surveying, and coordinating and will be summarized in this paragraph.

- **Organizational requirements** consist of requirements on the mapping of planning entities and on the mapping of decentralized coordination structures. It has to be mentioned that a planning entity does not necessarily has to correspond to an OEM or a supplier. It might be convenient to model a certain factory as a planning entity of its own. So far the needs of logistics service providers have not been examined in detail so that their mapping to planning entities is not yet supported. Agents shall represent planning entities rather than organizations in order to allow for greater flexibility if the organization has distributed production facilities. To enable participation at collaborative planning processes, the agent must have an understanding of communication processes and of its own situation and possibilities. Its situation comprises models of products and suppliers, resources and capacities, measures for the adjustment of capacities, demand, and time. Its possibilities are formed by a user defined set of plans of which it can choose appropriate ones on its own. This is done by an evaluation of its situation and the possible results of each plan. Also, the agent contains a formal model describing the semantics of the employed planning concepts. The latter is called the ontology model. Both ontology model and time model are different from the other models in a way that they support the data models, which are cross sectioned by them. Each model needs to refer to a time model to ensure consistency in the planning process. Other agents will presumably have a different time model, e.g. based on their factory calendar. The ontology-based model view has to describe the semantics of the data models and the communication concepts in order to enable semantic verification of incoming messages.

- **Several data mappings** are required to represent company and network relations in a supply chain and have been mentioned in the preceding bullet point. The product structure will be modeled according to the Odette DCP approach. It contains the levels of platform, model, feature packages, and parts. Feature packages are a combination of parts
that grant a function to the car. Validation functions support the test if certain feature packages can be chosen with a model and also if two feature packages can be combined. Part aggregations have to be linked to allow calculations from upper to lower level. The sourcing structure links parts and products to one or more suppliers or customers. These relations have to be further specified by the agreements made in the respective contracts. Agreements govern delivery dates, order sizes, and flexibility. Referring to ILIPT, a distinction of BTS and BTO suppliers is also made. Manufacturing resources are qualified by their identifier, their classification as a key resource, and their capacity. The key resource concept comes from ILIPT and is the basis of a two-step capacity planning where in the first step only key resources are checked. Capacity information contains the following properties: capacity per time unit including means of specifying uncertainty, the results of adjustment measures, tolerances, minimum and maximum production volumes regarding models and feature packages. Adjustment measures are mapped in a distinct model. This model provides the agent with the possibility to access measures as plans, i.e. to make use of them. Model properties are: affected resources or sourcing structures, lead time, costs, availability within a certain period of time, and effect. The demand constraint model confronts demands with capacities. It records incoming orders and keeps track of resource utilization, consequently being a monitor for shortages. In order to provide this functionality, it has to set up relations that link incoming orders to outgoing orders with restrictions in between. Incoming orders will partially consist of forecasts. Their stochastic distribution must be included.

- The need for a common understanding between planning entities despite of their inner concepts has been derived from the LiNet project in Section 3. The implication for agents is that they must be able to support communication processes by translating between the incoming messages and the events to raise internally as well as the other way round. Moreover, conversations have to be kept track of because more than one conversation will be held at a time.

- The execution of processes for planning, surveying, and coordinating have different requirements. First, the goal-oriented choice of plans has to be assisted by a goal model that guides the agent when he needs to decide how to handle a situation. This leads to the next requirement, the combination of communication, data models, plan execution and goal orientation. The agent has to know when to execute which plan. The point in time is marked by incoming messages, the choice is made according to the information stored in the data models and the goals. Once the decision is made to execute a plan, it must be accessible in an executable format. Surveillance and coordination functions are contained in decentralized entities like the virtual order bank as mentioned in the ILIPT project. They must exhibit the same functions as described for the planning entities.

4.2 Non-Functional Requirements

As the framework is designed to offer great flexibility in the modeling of domain specific agent-based systems, it has to be able to be adapted to different applications itself. This can be achieved by the dimensions of modularity, extensibility, and connectivity.

- The scenarios in which the framework is used can differ widely. Certain building blocks might not be required in some scenarios whereas specialized data structures might miss in others. Furthermore, the existing implementation of some module might have to be replaced by one that fits the project’s needs better. In any case, the components of the framework containing data structures and planning processes have to be exchangeable and independent to use in order to meet the intention of providing a generic toolset, hence the request for modularity.

- Extensibility is very important because of the high dynamics and heterogeneity of automobile supply networks. Not every detail can be implemented and not every development can be anticipated. The conclusion is that extensions will become necessary in the future. Thus, a proper formal definition of concepts is essential. On the technical level, the demand is to use standard technology.

- The afore mentioned heterogeneity calls for connectivity to existing ERP and planning systems. As the framework does not target at replacing backend systems, it has offer interfaces that enable the use of any common storages of master data and transaction data.

5. Architecture and Technologies

It was postulated that every planning entity and the decentralized coordination structures be represented by an agent. To fulfill this requirement, two agent types will be used: Planning Agent and Coordination Agent. Planning Agents are the representation of planning entities, i.e. those divisions of a company that engage in planning processes on their own. From the
network’s point of view, this agent type ensures planning authority and information security. The Coordination Agent maps independent structures, e.g. a virtual order bank. It also executes tasks like running auctions which involve a neutral party.

Both the Planning Agent and the Coordination Agent are realized with the Jadex BDI-Agent System [10]. That means that agents in the framework are represented by Beliefs, Desires, and Intentions. This is a common way of modeling agents [11]. The architecture of an agent is depicted in Figure 1. There are three deviations from standard Jadex agents. First, an OWL-P-based Communication Module is established as a belief [12]. Second, plans refer to a sophisticated set of models which represent structural domain knowledge and are also embedded as beliefs. Models are designed with the ontology editor Protégé and compiled to Java classes. As such, they can be accessed from plans. Last, the Jena framework is introduced. Jena is an API for the Resource Description Framework (RDF). Its purpose here is to implement the ontology model. The architecture is depicted in figure 1.

They are modeled as ontologies with Protégé and transferred to Java classes. Java classes can easily be integrated in agents, which in fact consist of a set of Java classes referenced in the agent definition file. The language used for ontologies is the Web Ontology Language (OWL) which is a standard set by the W3C. There are three variants of this language. OWL DL was chosen for the framework.

- The **Sourcing Model** (cf. Figure 2) gives information about the parts sold to a customer and procured from a supplier as well as the bilateral contracts made with each partner in the supply chain. Contracts encompass information on volumes and accepted deviations, term time, agreements on flexibility, and more. In order to create a connection from the customer to the supplier, products have to be split up in their constituent parts. The BOM link contains factors for their calculation. Two different ways to model BOM links are presented here. OEMs often specify their products differently in different time horizons. They employ a calculation based on (car) models for long-term planning without traversing the BOM down to parts. A model consists of options which are specified via an option mix, e.g. 20% of this car model will be ordered with a sunroof. Options can be resolved to parts and can additionally force some other parts to be included or excluded.

- The **Resource Model** contains all manufacturing resources that are relevant for planning (machines, workers, etc.) including their capacities. Regarding flexibility, this model also describes uncertainty concerning availability.

- The **Adjustment Measure Model** provides the necessary structures to represent network adaptivity. Network adaptivity also refers to local manufacturing and reflects in the division of
measures into resource measures and network measures. They are identical except that the resource measures reference local resources whereas the network measures reference supply relationships. The rest are common attributes that describe measures by lead-time, effect, costs, availability, external approval, and pursued goal. The latter establishes a relationship to the Entity Goal Model as it allows choosing that measure with supports the current goal best.

- The Demand Constraint Model is the implementation of the entity’s capacity and demand planning. It consists of three layers. Incoming demand is mapped to the Incoming Demand Layer where it is aggregated with existing demands. The Constraint Layer brings together demand and capacity. The Outgoing Demand Layer contains the demand that is forwarded to the suppliers. If a capacity shortage was detected, a plan to collaboratively solve this problem can be instantiated. The Coordination Agent is set up a little differently. It includes the Network Constraint Model and the Supply Demand Model. The former is like a Sourcing Model for the entire network and contains all supply relations between entities. No internal planning data is stored in this model. The latter is constructed in analogy to the Demand Constraint Model and maps the development of supply and demand on the network level.

- The Time Model is a three-dimensional calendar that allows converting the cooperative understanding of time, the factory calendar, and the Gregorian calendar. Each other model may reference the Time Model, e.g. to fix dates.

- The Ontology Model contains the semantics of the concepts used in the planning system. It is able to check if incoming messages are consistent with the agent’s concepts.

- The Goal Model provided by Jadex is sufficient for the framework. Its four types Perform, Achieve, Maintain, and Query are useful for collaborative planning and do not need to be extended.

- The Communication Module contains the interaction protocols which were modeled in OWL-P. As it has to be available during the lifetime of an agent, it is included as a service plan that handles all incoming messages. It is also the receiver of all internal messages that have to be send to other agents. Being the message bottleneck, it can serve the desired function of connecting external and internal information flows.

It was demanded that plans have to be executable, which means that they must be implemented as Java classes. Jadex provides a standard plan which can be extended. The user can then work with the framework specific plan which has methods and objects already implemented.

6. Prototypical Application

The scenario for the prototype is taken from the ILIPT project. Starting point is the arrival of a new demand forecast at the OEM. The network planning process will then distribute demand to the assembly factories. The demand for parts from suppliers can then be calculated and transmitted to the virtual order bank. After saving this data for surveying agreements on capacities and deliveries, it is forwarded to the suppliers which in turn do their own material planning and forward demands to their suppliers. The prototype contains only one supplier. In all, there are three agents: a planning agent for the OEM (network planning, requirements explosion, forwarding processes), a coordination agent for the VOB (saving agreements, raising alerts), and a planning agent for the supplier.

This simple scenario required the definition of protocols and plans. AUML [13] was used to formalize communication processes. They were subsequently transcribed to XML. The planning phases mentioned in the protocol now had to be implemented as plans for the agents. “forecasthandling” and “handledemandupdate” both extend the standard plan. The last step was to formulate the agent definition file. The communication module was included as a belief and its service plan “communicator” was registered. The protocol base and the domain models were also integrated as beliefs. Whereas these parts were added to both agents, the plans were assigned solely to the coordination agent and the planning agent. The test was executed in the Jadex Control Center. After reading the ADF files and starting the agents, an initial “NewForecast” message was sent. The resulting conversation is depicted in Figure 3.

Concerning the work the user has to do, it can be observed that many activities typical for agent implementations are facilitated by the framework. Most domain concepts are included in the framework models so that plans can be derived from framework classes.
The prototype can demonstrate the general applicability of the framework. Nonetheless, it is not a complete implementation of the concepts described in this paper. The communication module in the prototype is not based on OWL-P and a workaround is in place at this time.

7. Conclusion and Outlook

We have shown that our proposed framework can alleviate the work of implementing an agent-based system that operates in the field of supply chain management. Work concerned with agent concepts has been reduced. Now, the main task is to define plans, to connect to data resources, and to formalize interaction protocols. A binding to a special setting does not exist; all conceivable supply chain scenarios can be modeled. That way, concepts of agents and concepts of the domain of supply chain management have been brought together more closely.

The amount of technologies involved makes the development process a little difficult to understand. The next step will be to formulate a methodology that gives detailed instructions of how to use which tool to come up with a solution.

Further research regarding goals is needed. The strategy of an organization has influence on its plans, but is not explicitly modeled. Rather, it is implicitly contained in the business processes. It is difficult if not impossible to represent long term strategic plans in a way that they mildly influence the choice of plans, as the agent will choose that plan that solves its current problem best.

Apart from technical aspects which were discussed in this paper, other issues have to be handled before multi-agent systems will gain acceptance in supply chain management. Among those are for example the question of liability for agent transactions or IT safety concerns. However, we are confident that agent technology will become more important in the field of supply chain management in the future and that a framework for development is crucial for its success.

8. References


[8] F. Gehr, B. Hellingrath (Eds.), Logistik in der Automobilindustrie - Innovatives Supply Chain Management für wettbewerbsfähige Zulieferstrukturen, Springer Verlag, 2006


[10] A. Pokahr, L. Braubach, W. Lamersdorf, Jadex: A BDI-Agent System Combining Middleware are Reasoning,

Software Agent-Based Application, Platforms, and Development Kits, Birkhäuser, Basel, 2006

