Agent-based Collaborative Supply Net Management

Bernd Scholz-Reiter and Hartmut Höhns
University of Bremen, Faculty of Production Technology, Department of Planning and Control of Production Systems, P.O. Box 33 05 60, D-28335 Bremen, Germany
Email: {bsr, hoe}@biba.uni-bremen.de

Abstract

Supply chain management means the coordination of all material and information flows throughout the entire value chain. The main goal is to organise the overall process of the supply chain to achieve an optimum in costs and time, for example, by improving the following four Figures, such as increasing capacity utilization, decreasing inventories, decreasing lead time and increasing delivery reliability and adherence to delivery dates. The conventional control of supply chains pursuit top-down planning approaches to coordinate the supply flows (incl. material and information), run by current ERP systems, between the different enterprises participating in a supply network. Collaborative supply net management has technological and non-technological aspects. From the non-technological point, aside from willingness to organisational changes, a feeling of trustful partnership must evolve. On the technological side, agent technology is proposed as being indispensable to achieve integration of Advanced Planning Systems (APS) across company borders. Thus, this paper proposes an agent-based approach to collaborative supply net management, based on the SCOR-Model (Supply chain Operations Reference-model). Finally an example is given that sketches the idea of an agent-based simulation approach to tackle the bullwhip effect.

Keywords

Collaborative Supply Chain Management, multi-agent systems, SCOR-Model.

INTRODUCTION

The coordination of all material and information flows throughout the entire value chain is the focal point within supply chain management. To realise the overall coordination modern information technology (IT) is the essential facilitator and accelerator. At the present time the different participants within a supply network try to coordinate the information and material flows by deploying their existing IT. This very often leads to a heterogeneous IT landscape, basically comprising different types and brands of enterprise resource planning (ERP) systems up-to-date advanced planning systems (APS). Currently the participants align these IT systems to each other, by using interfaces for purpose of communicating through electronic data interchange (EDI) standards like EDIFACT or ODETTE. First attempts have already been made to substitute these interfaces and standards by using more flexible meta-languages like the extensible markup language (XML), for data exchange and

H. S. Jagdev et al. (eds.), *Collaborative Systems for Production Management*

integration purposes [1]. By using common ERP systems, for the coordination purposes within supply chain management, often problems arise, because these systems basically pursue a multistage planning process [2]. This means, that all material requirements and capacities are aligned one after another and not parallel at the same time, without allowing for revisions to upper-level decisions [1, 2]. Furthermore, the optimisation approach basically focuses and optimising all logistical sub-systems within one enterprise, which leads to a purely internal approach [2]. Some tasks like the bill-of-materials processing (BOMP), do not consider capacities all [1]. Often lead-times are used as a fixed input for the BOMP, while they should be a result of planning [1]. The occurring problems within the intercompany information flow and the related problems within the intercompany material flow, as a result of the use of different, common ERP systems, are summarised within Figure 1.

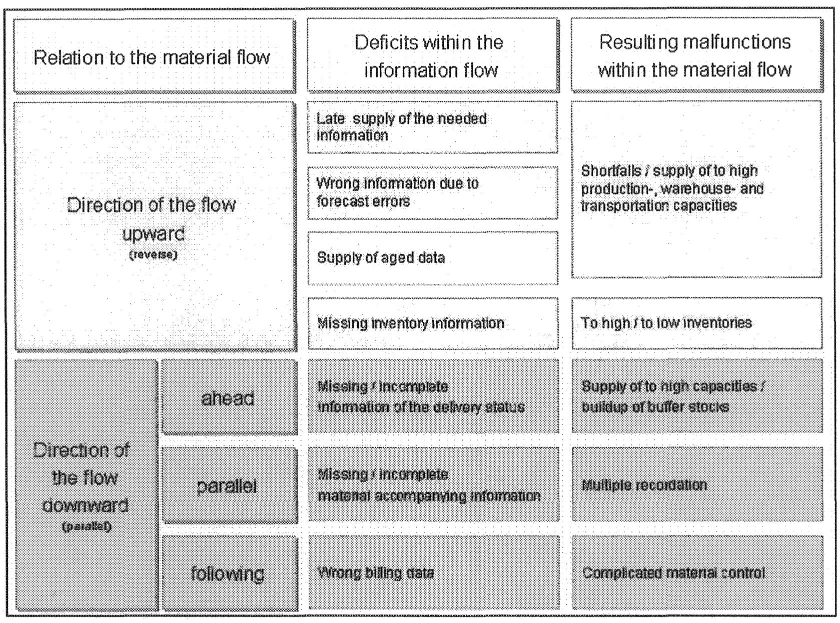

Figure 1 - Malfunctions within the information flow and the effects on the material flow caused by conventional ERP-Systems [3]

The up-to-date APS systems do not substitute but supplement the ERP systems [1]. They emanate from the operative transaction units of the common ERP modules [2], and are intended to remedy the defects of ERP systems [1]. APS systems take over several the planning and scheduling tasks, for the purpose to simultaneously align all relevant parameters within a supply chain, especially constrains and bottlenecks, while the ERP system is still required as a transaction and execution system e.g. for orders [1, 2]. The APS planning approach is based on a model of the actual supply chain, which can be run by a simulator [1, 2]. During the simulation the currently known internal and external constrains and bottlenecks are projected onto the model of the supply chain, according to the actual orders. As a result, some alternative proposals are given ranked by their potentially most benefit [1, 2]. The question which arises form the APS approach is, if the considered and deployed information, in this case given internal and external constrains and bottlenecks, are still valid after a period of time, to serve as input for model-based planning.

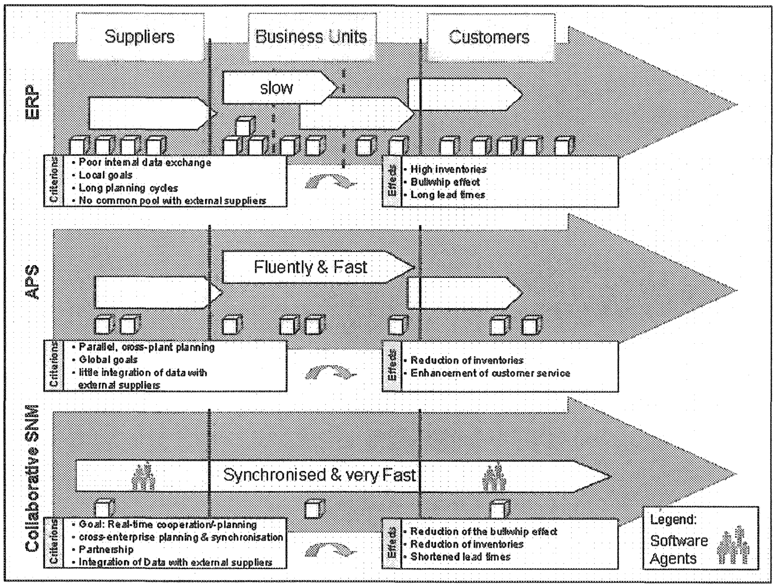

Figure 2 - The basic differences between the ERP, APS and collaborative SCM approach according to [4]

In contrast to the ERP and APS (Figure 2), this paper describes an agent-based, respectively a multi-agent approach to a more heterarchical approach to supply chain management. A major task will be to define and derive the needed, different types of software agents, regarded as software artefacts or intelligent software modules, from redefined cross-enterprise business processes. This is aimed to lead to a collaborative supply net management approach, through enabling and supporting cross-enterprise, business process oriented interoperability [4].

The main tasks of the different types of software agents will be the timely coordination of information flows, monitoring and triggering issues and as well the execution of suitable sub-processes in order to reduce inventories, lead times and therefore increase delivery reliability [4, 5]. Basically the idea is to just *simply* substitute inventories by fast just in time information (JIT), at the right time, the right place and the right quality, which already points out the major challenge behind this idea [4]. Within this paper the supply chain management approach is based on the SCOR-Model, which will, on the one hand, partly serve for the redefinition of the cross-enterprise business processes, whereas on the other hand it will provide a good and widely agreed structure and content, which should support the definition and derivation of the software agents.

REACTIVE PLANNING AND CONTROL AND CROSS-ENTERPRISE INTEROPERABILITY

The software agent concept seems to be a very promising approach for planning and control activities within the field of logistics and especially concerning cross-enterprise coordination and cooperation tasks within supply chain management. Concerning the use within the field of short-term planning and control of logistics processes, software agents are expected to be able to cope with the existing dynamics. The Figure 3 arranges the software agents within the aspired field of application, in order to support a business process oriented interoperability.

The approach of reactive planning and, in addition, reactive control is regarded as an important completion to the short-term planning. The actual plans become invalid if unforeseen disturbances occur, e.g. deliveries delays or even failure of deliveries, machine break downs or short-term changes of customer wishes. Reactive planning and control means a continuous extraction of information about the environment, out of the current running logistics processes. The gathered information needs to be processed in an appropriate way. The aim is to draw conclusions which will be used immediately to control the running process [6]. Usually, there is no knowledge in advance available about the type of the emerging disturbances and the affected point within the logistic business process chains. The conventional approach of short-term planning and control therefore has to fail because there is no complete information available. The extracted information from the environment change again in the mean time and therefore cannot be taken into account for the short-term planning and control. Therefore, the extraction and processing of the information as well as the drawing and output of conclusions need a continuous interaction with the environment. The following Figure 4 shows a general hybrid design concept of an intelligent agent according to [7], which so far has not been implemented in any available agent development toolkit.

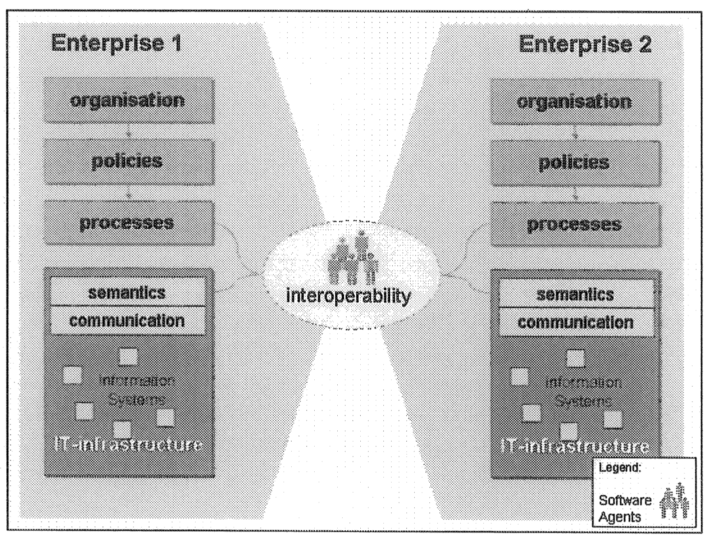

Figure 3 - Agent supported cross-enterprise interoperability

Within Figure 4 the software agent is embedded within the running business processes through performing a set of sub-processes. Through sensors and actors (out put units) it is enabled to perceive changes within its environment. The software agent compares the changes with existing patterns and tries to find a matching plan from its library within the competence module. If the software agent does not find any executable and matching plan, it starts a re-scheduling process (involving the planner, scheduler and reasoner components) based upon its knowledge and the given goals. Finally the re-scheduled tasks will be executed as an activity and communicated to the concerned software agent *neighbour*. Thus it is expected, that a software agent will be able perform these task related re-scheduling, planning, searching and executing actions faster than a human or semi-automatic operated cross-enterprise ERP or APS system-based business process.

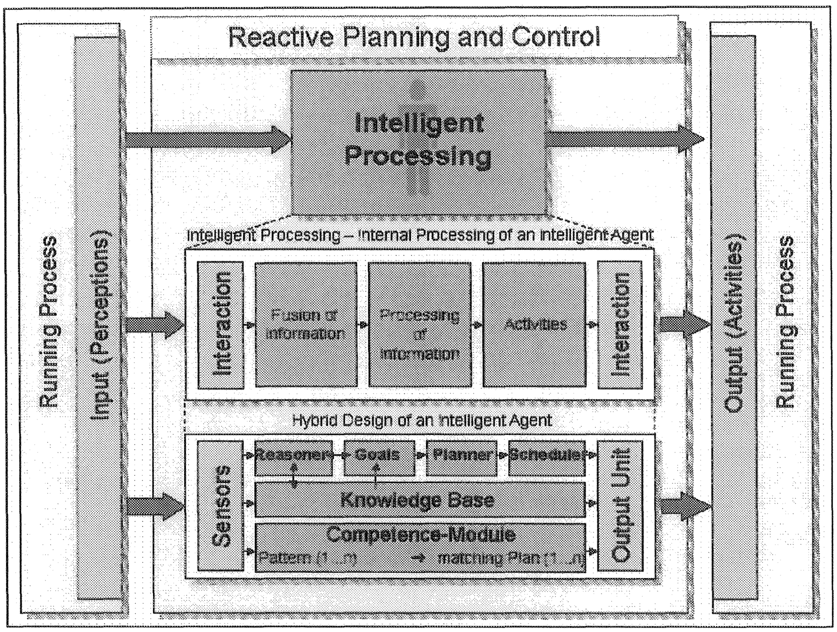

Figure 4 - Agent-based reactive planning and control adapted from [7]

SOFTWARE AGENTS AND MULTI-AGENT SYSTEMS

Currently agent technologies are on their way from academic research to practice. But there is still a need for research and testing to get agent technologies into use for different scenarios. Up to now, it is still very difficult to give a comprehensive definition of what a software agent really is. Initially, one has to distinguish between software agents, intelligent or autonomous agents and multi-agent systems. Secondly, the appearance and performance of the agent seems to depend very much on the different implemented and deployed development tools. A short and probably widely accepted definition would be the following according to [8]: A software agent is a software programme, which runs for longer periods and can be described as an autonomous dispatch of tasks or pursuit of goals in interaction with its environment.

This definition includes the intelligent or autonomous agents by the notion of autonomous respectively autonomy, which, according to Jennings and Wooldridge [9], is a very difficult concept to realise. Basically the concept of autonomy just tries to express that the agent, regarded as a system, should be able to act without the direct interventions of humans [9]. Thus the agent has control over its own actions and internal states [9]. This directly leads to the term of intelligence, as the major challenge from the development point of view [9].

Multi-agent systems always emerge when several more or less autonomous and heterogeneous agents act together as a loose coupled network, to cooperatively solve a given problem [10]. Thus the resulting characteristics of multi-agent systems can be considered as follows [10]:

- Each agent has incomplete information, or capabilities for solving the problem, thus each agent has a limited viewpoint.
- There is no global system control.
- Data is decentralised.
- Computation is asynchronous.

Finally, all definitions of agent technology can be summarized by the following statements, which can easily act as guidelines for the development and application of suitable agents [9, 11]:

- Agents are a powerful, natural metaphor for conceptualising, designing, and implementing many complex, distributed applications.
- Agent systems typically use AI techniques – in this sense, they are an application of AI technology – but their "intelligent" capabilities are limited by the AI's state of the art.
- Development techniques for agent systems are in their infancy. Thus it often is observable, that the development of any agent system, however trivial, is essentially a process of experimentation. Unfortunately, the experimental process encourages developers to forget that they are actually developing software. Mundane software engineering processes, like requirements analysis, specification, design, verification and testing, are easily forgotten and the project slows down and stagnates. Thus developers can first of all adapt object-oriented techniques to great effect.
- It has to be considered as common to all agent-based applications, that they are no overall system controller and have no global perspective, already by definition.

This understanding and definition of agent systems must inherently have an effect and lead to several possible perspectives on the organisation and modelling of these systems, which can have either a theoretical, phenomenological or technological orientation [12].

This section of the paper will only focus on the organisation and modelling of multi-agent systems, because multi-agent systems are an important paradigm for building complex and distributed information systems, especially cooperative ones, in information-rich environments [13]. Information-rich environments comprise the following characteristics according to Huhns and Singh [13]:

- They span enterprise boundaries.
- They include heterogeneous components.
- They comprise information resources that can be added or removed in a loosely structured manner.
- They lack global control of the accuracy of the contents of those resources.
- They incorporate intricate interdependencies among their components.

As a result, Huhns and Singh suggest that cooperative information systems can be regarded as quasi inherently multi-agent systems with organizational and database abstractions, that are furthermore geared to open environments [13]. Thus they propose a standard set of agent types for the development and deployment of cooperative information systems or multi-agent systems. This standard set of agent types has been aligned to SCOR-Model within the Figure 5 and can be defined as follows [13]:

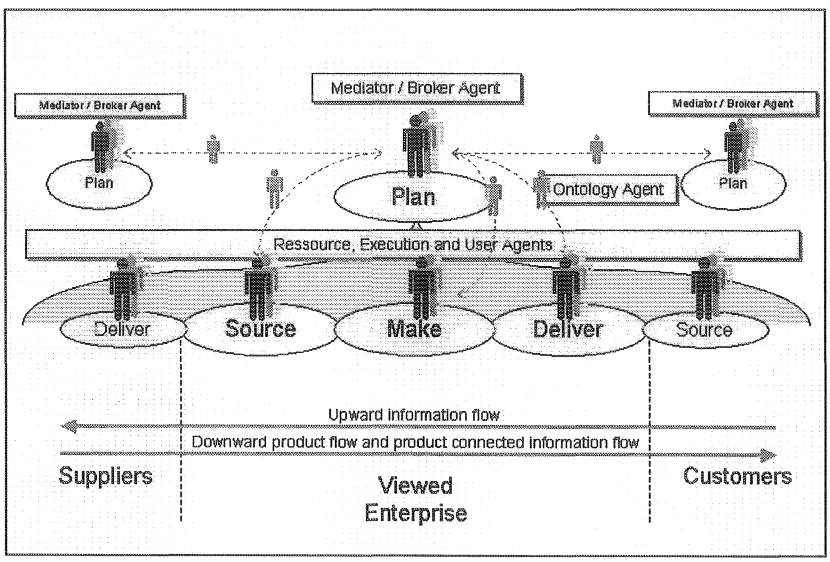

Figure 5 - Agent-based supply net management based on the SCOR-Model

- The **user agents**, for example provide access to other information resources, such as data analysis tools or workflows.
- The **broker agents**, for example implement a yellow and white pages directory service for the purpose of locating other agents who have the needed and appropriate capabilities.
- The **resource agents** can appear as wrappers. They implement common communication protocols and translate them into and from local access languages. They can as well have several other functions, depending on the resource, which it represents.

- The **execution agents**, for example are able to execute workflows, which might extend over the web. They are mostly implemented as a rule-based knowledge system.
- **Mediators** are considered specialized execution agents. They combine, for example, partial responses that have been obtained from multiple resources or determine appropriate resources that might have relevant information by using help from brokers.
- The **ontology agents** are essential for interoperation, for example by providing a common context as semantic grounding, that agents can then use to relate their individual terminologies.

Based on these definitions of a standard set of agent types, Huhns and Singh assume that most of the agent-based information systems will incorporate one or more of these basic types.

COLLABORATIVE SUPPLY NET MANAGEMENT

This section of this paper introduces the agent-based collaborative supply net management concept based on the SCOR-Model, which is aimed to lead to a business oriented definition and derivation of the different agents in contrast to holonic approaches according to [14]. According to [15], collaborative business has both technological and non-technological aspects. From the non-technological point, aside from willingness to organizational changes, a feeling of trustful partnership must evolve [15]. On the technological basically three essential techniques are foreseen [15]. Thus according to [15] agent technology is proposed on the one hand, and on the other hand it is indispensable to achieve integration of Advanced Planning Systems (APS) across company borders, to install an inter-company supply net logistic control and finally to deploy a virtual private network (VPN) based planning data exchange. Undoubtedly software agents represent an essential technology to shape and enable collaborative business.

Based on the SCOR-Model and its current version 5.0 [16], a four level business process modelling approach is proposed and displayed in Figure 6. Beside the SCOR core management processes, like source, make, deliver, return and plan, the SCOR-Model distinguishes between the following three fundamental process types [16]:

- **Planning**: A process that aligns expected resources to meet expected demand requirements. Planning processes: Balance aggregated demand and supply, Consider consistent planning horizon, (generally) occur at regular, periodic intervals can contribute to supply chain response time.
- **Execution**: A process triggered by planned or actual demand that changes the state of material goods. Execution processes: Generally involve scheduling/sequencing, transforming product, and/or moving product to the next process and can contribute to the order fulfilment cycle time.
- **Enable**: A process that prepares, maintains, or manages information or relationships on which planning and execution processes rely.

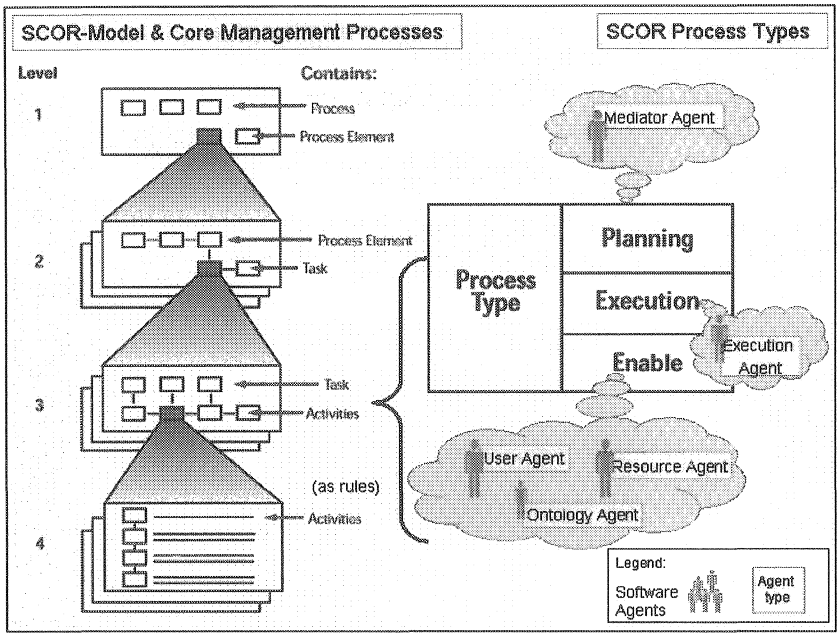

Figure 6 - Deriving the cooperative software agents from the SCOR-Model

The introduced idea, for the identification of the software agents, sets in from the level three of the SCOR-Model. At this level, top-level tasks and the first activities have already been aligned. This will provide a reliable basis to start from in order to identify, derive and describe suitable software agents, while considering the described SCOR-Process Types and the standard set of agents as proposed by Huhns and Singh (Figure 6). Thus in a first attempt the proposed set of standard agents has been matched to the SCOR Process Types (Figure 6). Some of the already described activities within the SCOR-Model can be deployed as services, in the sense of skills, provided by the agents. Furthermore, the so far described activities point out an appropriate direction to define suitable decision rules.

This first sketch of a SCOR-Model based approach is aimed support the development of agent-based systems respectively multi-agent systems, addressing all kinds of cross-enterprise, collaborative planning processes which are according [4]:

- forecast collaboration,
- capacity collaboration,
- order collaboration,
- inventory collaboration, and
- transportation collaboration.

ORDER COLLABORATION

This section addresses the topic order collaboration and the related bullwhip effect, as a consequence of a fragmented supply chain and the absence of collaborative reconciliation of the participants [4]. The bullwhip effect describes the increasing amplification of orders occurring within a supply chain the more one moves upstream [1, 17]. Surprisingly, this phenomenon also occurs even if end item demand is fairly stable [1, 17]. Based on monthly batch runs and due to slow computer systems, high inventories and a fairly low amount of variants, up to now this effect was disguised in practice and basically not further analysed. The bullwhip effect is a problem of the order fulfilment process, which includes according to [18] the supply chain processes like procurement, order processing, production and distribution. Thus the objective of supply chain management is to optimise the order fulfilment process by coordinating the supply chain activities and streamlining the transition of intermediate products from enterprise to enterprise via efficient logistics management [18].

The causes of insufficient cross-enterprise coordination and reconciliation can be summarized as followed according to [4]:

- Absence of transparency concerning current requirements and offers directly to the customer or along several ranges of the supply chain up to the end customer.
- Time delays caused by insufficient coordination of planning cycles.
- No timely cross-enterprise detection and consideration of material and capacity constrains and bottlenecks.
- No feedback (loops) and often not processable information concerning constrains and bottlenecks to predecessors and successors during the planning process.
- Media break (fragmented information chain) and manual processing, as well as missing logical cross-enterprise integration of existing planning systems (IT).

The following 7 addresses the bullwhip effect as a problem of the order fulfilment process. This represents a further development of the proposed agent-based reactive planning and control of logistics processes approach in [19]. It is aimed to forward the right parts of an order (incl. an experimental bill explosion possibly supported by a product setup (user agent, core process: deliver), to the right supplier immediately, while monitoring the awaited deliveries (resource agent, core process: source), the current pipeline stock (resource agent, core process: source), meaning the goods that have just been taken in, as well as the current stock (resource agent, core process: make).

Furthermore the aim is to choose a suitable ordering policy and to decide whether to release an order to a supplier or not (execution agent, core process: source). As a result it is expected to prevent inventory buildup and to continuously readapt the safety stock (mediator agent, core process: make), while considering and partly adapting the proposed theorems and equations for order balancing and storage control from [20, 21, 22]. The basic idea of the approach has already been implemented conventionally and achieved good results, as supplement system to common ERP systems, without using software agents and for the only purpose of inventory control [22]. This still leaves open the question, how the calculated results should be communicated through out the supply chain and how a transparent order collaboration process will be realised. Thus the basic idea has been picked up to try out an agent-based approach to collaboratively manage the order flow and strongly related inventories. In Figure 7, the deployed types of agents are matched to the suggested set of standard agent types according to Huhns and Singh [13].

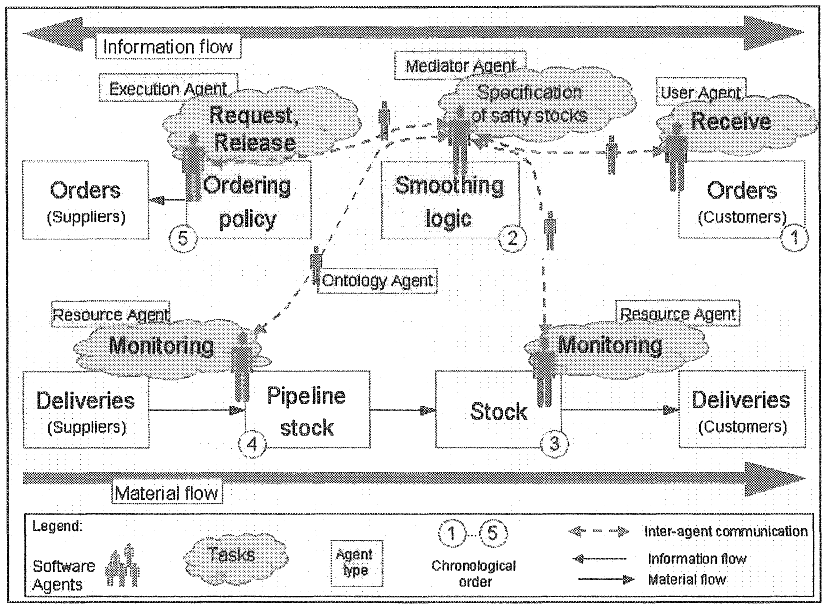

Figure 7 - Cooperative software agents are tackling the bullwhip effect

The adaptation of the proposed price delivery relations, according to [23], will provide a very useful negotiation medium to support the agent-based cross-enterprise interaction. At the present time this proposed price delivery relation has been designed and proved just for internal use only aimed at efficient allocation of manufacturing facilities and thus to prevent from queuing problems [23]. But the basic idea appears to very suitable to serve as cross-enterprise negotiation medium, after the design of some extensions (e.g. contract penalty, procurement costs). The suitable supplier will be appointed based on the ability to deliver in due time and on the quoted price (high price – short delivery time and vice versa). The basic idea is that, in this case, the execution agent is running a few negotiations with relevant suppliers at the same time, in order to initially request a product and finally release an appropriate order and thus appoint a suitable supplier. The negotiations will be carried out on an cross-enterprise marketplace from the field of e-commerce (e.g. Covisint) supported by software agents to achieve interoperability.

Such an cross-enterprise marketplace is determined by the following features adapted from [23]:

- 2 to 4 negotiation partners in the sense of a demander and a contractor,

- A negotiation package in the sense of a service bundle or a specified, requested product,
- Negotiation parameters, in this case a relation of price and delivery date, in the sense of available to promise (ATP) considering the delivery date,
- And a negotiation space, in which it is possible to match the request and the offer, in the sense of a tolerable latest delivery date or highest price.

After this short introduction of the agent-based concept to order collaboration, it is finally aimed to achieve an agent-based simulation framework to try out this concept and to analyse its impact on a model of a supply net. To realise this task, real datasets have been inquired of cooperative enterprises from the field of the manufacturing industry. A four layer model, as shown in Figure 8, serves as basis for the scientific research approach to realise the agent-based simulation framework, and displays the different steps to be taken.

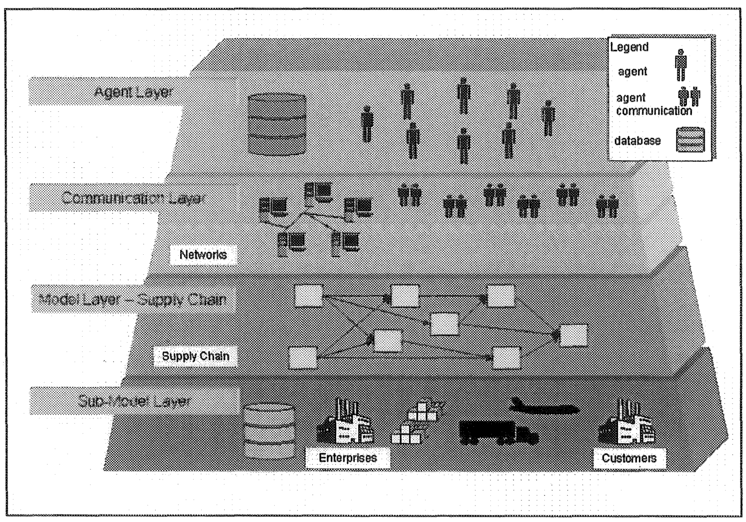

Figure 8 - The four layer model as the basis of the scientific approach

- **Step1 – Modelling of the properties**: The different properties (e.g. transportation vehicles) of the participating enterprises and customers are represented bottom-up within the sub-model layer. First of all these properties, as elementary parts of the model of the supply net, have to be modelled. Therefore all necessary parameter of the properties need to be considered (e.g. current stock on hand).
- **Step2 – Completion of the model of the supply net**: The bottom layer is followed by the supply net model layer, representing the enterprises and customers as knots of the supply net. These knots consist of the different properties, which already have been modelled within the sub-model layer. The knots will be linked by the flow of products from suppliers to manufactures to the customers.

So far a static model of the supply chain has been achieved. But the aim is to achieve a dynamic model of the supply chain, by deploying the set of software agents to control the knots (e.g. supplier, manufacturer, customer) and the linked flow of orders and products.

- **Step3 – Realising the communication layer**: The communication layer will be mainly focussing on the exchange of information between the applied agents and the knots of the supply chain. It links the supply chain model layer with agent top layer, in the sense of a communication interface. A network of different computers, which run the agents and the model of the supply chain, has to be built up. This also means defining the possible communication channels between the agents.
- **Step4 – Defining, deriving and implementing the software agents**: The agent layer itself represents the described set of cooperating software agents, applied to plan, execute and enable the major tasks of the supply chain and its logistics processes. The agents will be connected with the simulation tool via its provided interfaces. Thus the defined set of agents needs to be developed and equipped with an appropriate set of strategies to pursue their task specific goals.

CONCLUSIONS AND OUTLOOK

Currently the introduced agent-based collaborative supply net management is a concept, based upon the SCOR-Model. It still has to be tried out and evaluated through the described agent-based simulation framework. The purpose is first of all to qualitatively and quantitatively analyse and measure the bullwhip effect and to finally derive basic rules to control this effect, as the stated problem of the order fulfilment process. Nevertheless, this SCOR-based approach to identify, derive and describe suitable software agents seems to be very promising. Though it has not been evaluated yet, the followers of the SCOR-Model are already a large, well supported and fast growing community (Supply Chain Council). The user requirements and the trends in supply chain management, e.g. to business orientated cross-enterprise integration, aiming towards interoperability seem to support this approach (Supply Chain Council). Thus it is aimed to further develop this approach.

Maybe the use of an extended eCoReference Architecture [24], which is shown within Figure 9, is a possible completion of the so far described SCOR-based approach. Basically this reference architecture has been developed to model electronic (virtual) market places, by the CommerceNet, a US-Industry-Syndicate, between 1994 and 1999. The major benefit of this approach is that it clearly illustrates the modelling steps (top-down and bottom-up) as follows:

- Starting from the virtual cross-enterprise market,
- Followed by the business, in this case the SCOR Core Management Processes,
- Followed by the services (in the sense of skills) provided by the software agents,
- Followed by the software agents interaction as a derivation of the provided services, and finally,

- Followed by the related documents, in this case the interaction protocols and their content.

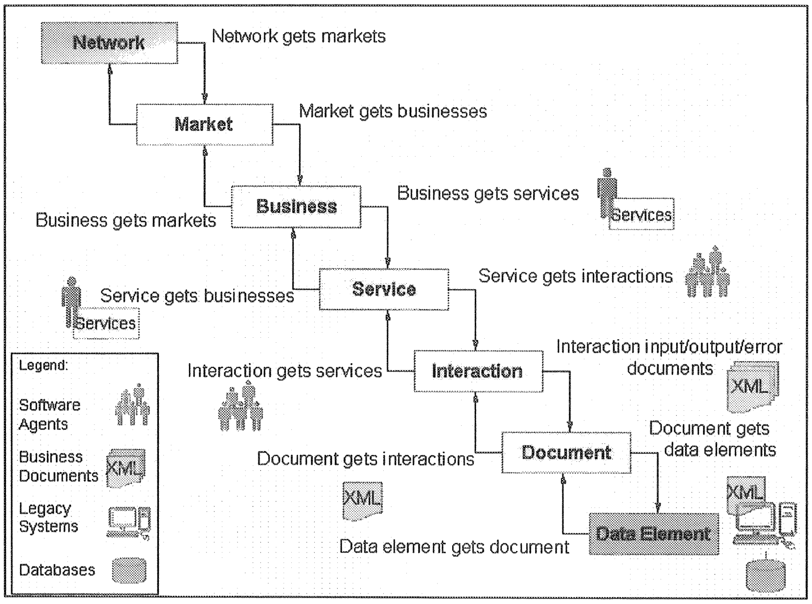

Figure 9 - Modelling collaboration deploying an extended eCo-Reference Architecture

Finally it can be used as an extension of the SCOR-approach, below the levels 3 and 4 referring to Figure 6. As an outlook on future work and according to [15] an essential technological aspect, the Figure 9 addresses the real world implementation of software agents, as a useful cross-enterprise integration technology, through coupling it to legacy (e.g. ERP/APS) systems.

ACKNOWLEDGEMENTS

The work presented in this paper is a part of a research project*, which is currently founded by the Deutsche Forschungsgemeinschaft (DFG), SCHO 540/8-1 (Department of Planning and Control of Production Systems, University of Bremen) in cooperation with the DAI-Lab at the Technical University of Berlin.

REFERENCES

[1] Stadtler, H., Kilger, C., 2000, Supply Chain Management and Advanced Planning, Concepts, Models, Software and Case Studies, Springer Verlag, Berlin Heidelberg.

[2] Werner, H., 2002, Elektronische Supply Chains (E-Supply Chain), In: Busch, A., Dangelmeier, W., 2002, Integriertes Supply Chain Management, Gabler, Wiesbaden: 403 - 420.

[3] Steven, M., Krüger, R., Tengler, S., 2000, Informationssysteme für das Supply Chain Management, PPS-Management, 5/2: 15 - 23.
[4] Langemann, T., 2002, Collaborative Supply Chain Management (CSCM), In: Busch, A., Dangelmeier, W., 2002, Integriertes Supply Chain Management, Gabler, Wiesbaden: 421 - 438.
[5] Rautenstrauch, T., 2002, SCM-Integration in heterarchischen Unternehmensnetzwerken, In: Busch, A., Dangelmeier, W., 2002, Integriertes Supply Chain Management, Gabler, Wiesbaden: 343 - 362.
[6] Scholz-Reiter, B., Scharke, H., 2000, Reaktive Planung, Industrie Management, 16/2: 21 - 26.
[7] Brenner, W., Zarnekow, R., Wittig, H., 1998, Intelligente Softwareagenten, Grundlagen und Anwendung, Springer Verlag, Berlin Heidelberg.
[8] Görz, G., Rollinger, C.-R., Schneeberger, J., 2000, Handbuch der Künstlichen Intelligenz, Oldenbourg Verlag, München Wien.
[9] Jennings N. R., Wooldridge, M. J., 1998, Applications of Intelligent Agents, In: Jennings N. R., Wooldridge, M. J., Agent Technology: Foundations, Applications, and Markets, 1998, Springer Verlag, Berlin, Heidelberg: 3 - 28.
[10] Jennings, N. R., Sycara, K., Wooldridge, M. J., 1998, A Roadmap to Agent Research and Development, Autonomous Agents and Multi-Agent Systems, 1/1: 7 - 38.
[11] Wooldridge, M. J., Jennings N. R., 1999, Software Engineering with Agents: Pitfalls and Pratfalls, IEEE Internet Computing, 3/3: 20 - 27.
[12] Gasser, L., 2001, Perspectives on Organizations in Multi-agent Systems, In: Luck, M., Marik, V., Stepankova, O., Trappl, R. (Eds.), 2001, Multi-Agent Systems and Applications, Springer Verlag, Berlin, Heidelberg: 1 - 16.
[13] Huhns, M. N., Singh, M. P., 1998, Multiagent Systems in Information-rich Environments, In: Klusch, M., Weiß, G. (Eds.), 1998, Cooperative Information Agents II, Springer Verlag, Berlin Heidelberg: 79 - 93.
[14] Fischer, K., Funk, P., Ruß, C., 2001, Specialised Agent Applications, In: Luck, M. *et al.* (Eds.): ACAI 2001, Springer Verlag, Berlin, Heidelberg: 365 - 382.
[15] Sundermeyer, K., 2001, Collaborative Supply Net Management, In: Baader, F., Brewka, G., and Eiter, T. (Eds.), 2001, KI 2001: Advances in Artificial Intelligence, Springer Verlag, Berlin, Heidelberg: 467 - 470.
[16] Supply-Chain Council, 2001, Supply chain Operations Reference-model, Overview of SCOR Version 5.0, from {www.supply-chain.org}, download: 21.06.2002.
[17] Forrester, J.W., 1961, Industrial Dynamics, Productivity, Cambridge Norwalk.
[18] Tan, G. W., Shaw, M. J., Fulkerson, B., 2000, Web-based Supply Chain Management, Information Systems Frontiers, 2/1: 41 - 55.
[19] Scholz-Reiter, B., Höhns, H., 2001, Reaktive Planung und Steuerung von logistischen Prozessen mit Multiagentensystemen, Industrie Management, 17/6: 33 - 36.
[20] Carlsson, C., Fuller, R., 2001, Reducing the bullwhip effect by the means of intelligent, soft computing methods, In: Proceedings of the 34th Hawaii Conference on System Sciences, 2001, IEEE.
[21] Arnold, D., Faißt, B., 2000, Implementation of an information exchange system within a supply chain – analysis of the bullwhip effect, In: Proceedings of the International Conference on Production Research, ICPR, Bangkok, 2000.
[22] Alicke, K., Kowalewski, M., 2001, Reduktion des Bullwhip-Effekts durch geglättete Auftragsdaten, In: Sebastian, H.-J., Grünert, T. (Hrsg.), 2001, Logistik Management, Supply Chain Management und e-Business, Teubner, Stuttgart, Leipzig, Wiesbaden: 83 - 91.
[23] Vollmer, L., 2000, Agentenbasiertes Auftragsmanagement mit Hilfe von Preis-Liefertermin-Relationen, VDI-Verlag, Düsseldorf.
[24] Merz, M., 2002, E-Commerce und E-Business: Marktmodelle, Anwendungen und Technologien, dpunkt-Verlag, Heidelberg.