The *Agent.Enterprise* Multi-Multi-Agent System

Jens Nimis[1], Tim Stockheim[2]

[1]Institute for Program Structures and Data Organization, Universität Karlsruhe (TH)
Am Fasanengarten 5, D-76131 Karlsruhe, Germany
nimis@ipd.uni-karlsruhe.de

[2]Institute for Information Systems, Johann Wolfgang Goethe-University Frankfurt
Mertonstr. 17, D-60325 Frankfurt am Main, Germany
stockheim@wiwi.uni-frankfurt.de

Abstract: This paper presents the development of the Agent.Enterprise system, which consists of five multi-agent systems from the manufacturing logistics domain. Consequently, the development process has to take the distributed structure of the involved projects into account. The maturity of the technical foundations for multi-agent systems and the support by development tools, infrastructure services and development methods leads to an increasing number of existing multi-agent systems and entails the need to couple them into large multi-multi-agent systems. The Agent.Enterprise development process combines aspects from established agent-oriented development with new concepts designed to interlink multi-agent systems. The structure of the coupled multi-agent systems is designed to inherently meet the requirements of distributed supply chains where information for integrated production planning and control is not available within the whole supply chain. This functionality is an integral part of the Agent.Enterprise System. As a consequence, the system is able to handle severe disturbances at supplier sites while dealing with highly customized and complex products.

Keywords: Multi-agent systems, Agent-oriented software engineering, Distributed planning

1 Introduction

Multi-agent systems (MAS) perfectly suit the demands for global flexibility, co-operation and at the same time local autonomy. Compared to existing SCM systems, the successful integration of numerous MAS that perform both inter- and intra-organizational planning and execution tasks is an innovation, which could lead to an improvement in the design of SCM systems.

The technical foundation for the development of large multi-multi-agent systems (MMAS) is provided by the growing success of the FIPA-standard [FIPA] for MAS and its compliant frameworks, e.g. JADE [BPR01], and the availability of an open service infrastructure, such as Agentcities [Agentcities]. Unfortunately, there is no guidance on how such systems should be built, since the existing agent-oriented development methods are focused on isolated MAS. Therefore, we present an approach for building MMAS that originate from the German priority research program 1083 “Intelligent Agents in Real-World Business Applications” where five projects from the manufacturing logistics domain integrate their MAS prototypes in one large MMAS called *Agent.Enterprise* [Frey03b].

Outline. Our *Agent.Enterprise* approach is presented in Section 2 together with the application domain and the Gateway-Agent Concept, which sets up some basic conditions for the approach. In Section 3 we show some characteristics of the resulting prototype. Finally, Section 4 draws some conclusions.

2 The *Agent.Enterprise* Approach

The *Agent.Enterprise* initiative is a platform which has as its goal to integrate recent research results and join forces in order to build up a networked, agent-based application scenario for the manufacturing domain [Frey03b]. In order to address the difficulties caused by the distributed structure of the involved projects, we developed a concept based on well-known methodologies of agent-oriented software engineering.

Over the last years, the need for applicable and broadly accepted development methods for multi-agent systems resulted in a large number of efforts undertaken to overcome this problem. Various methods exist, which support at least one of the established development phases (analysis, design, implementation, and deployment) with representations of varying formal accuracy and semantic foundations, e.g. Gaia [WJK00], PASSI [CP02], MASSIVE [Lind01], MaSE [WD01], AUML [OVB01].

The majority of the mentioned methods focuses exclusively on building a single (most often closed) MAS and thus, does not consider the development or integration of MMAS. Nevertheless, it can be expected, that the development process for MAS and MMAS will have some joint properties. The following subsections outline how we developed the *Agent.Enterprise* MMAS and present some underlying design decisions. (For a more detailed discussion of our approach and a comparison to some of the mentioned methods see [Sto04].)

2.1 The Development of *Agent.Enterprise*

Unlike GAIA, PASSI, MASSIVE and other agent-oriented development methods the *Agent.Enterprise* concept focuses on a distributed and weakly coupled development process, while minimizing the time required for face-to-face communication. Consequently, the initial design period is comparatively short and restricted to the constitution of the speech acts and the interaction protocol design.

The results of the analysis and design process are consolidated in functionally restricted prototypes, which constitute a test bed for the components of the evolving MMAS. The projects substitute their prototypes with gateway-agents in order to connect their applications to the common scenario. This requires a process of repeated cycles of redesign, implementation, and tests. Figure 2.1 depicts our development approach, while a detailed description can be found in the following subsections.

Role Definition and Assignment

The focus of the related research was taken as the major criterion to assign a specific functionality within the supply chain to the various participating projects. The roles assumed by a project will now be assigned in respect to its functionality. Yet, the next step is to bring life to the roles.

Use Case Specification

A first approximation of the use case specification is made by a simple role-playing technique. One member of the scientific staff of each participating projects takes over the role of her MAS and writes down its informational requirements. Following this, cards are handed out. The sender writes down the contents of the message as well as the receiver, therefore each card represents a single act of communication. Starting with the initiator of an order, in other words the customer, the whole supply chain is acted out until finally the last card announces the delivery of the order from the OEM to the customer. As a result of this role-playing technique, the communication acts between the projects' MAS as well as the required information are specified informally and have to be formalized.

Speech Act Design

After defining the roles for each participating MAS, it is necessary to ensure that high-level communication between the systems is possible. However, a language barrier for the communication exists due to the heterogeneity of knowledge representation and semantics in the individual systems. Consequently, we introduce the so-called Gateway-Agent Concept, which is outlined in the next subsection. This concept defines a virtual MAS where the agents are scattered across a number of agent platforms such that an ontology can specify the semantics of the conversations. While using ontological expressions as a means of communication, we chose to agree on a shared ontology for the communication between the gateway-agents, while future work may include semantic mediation based on a common terminology.

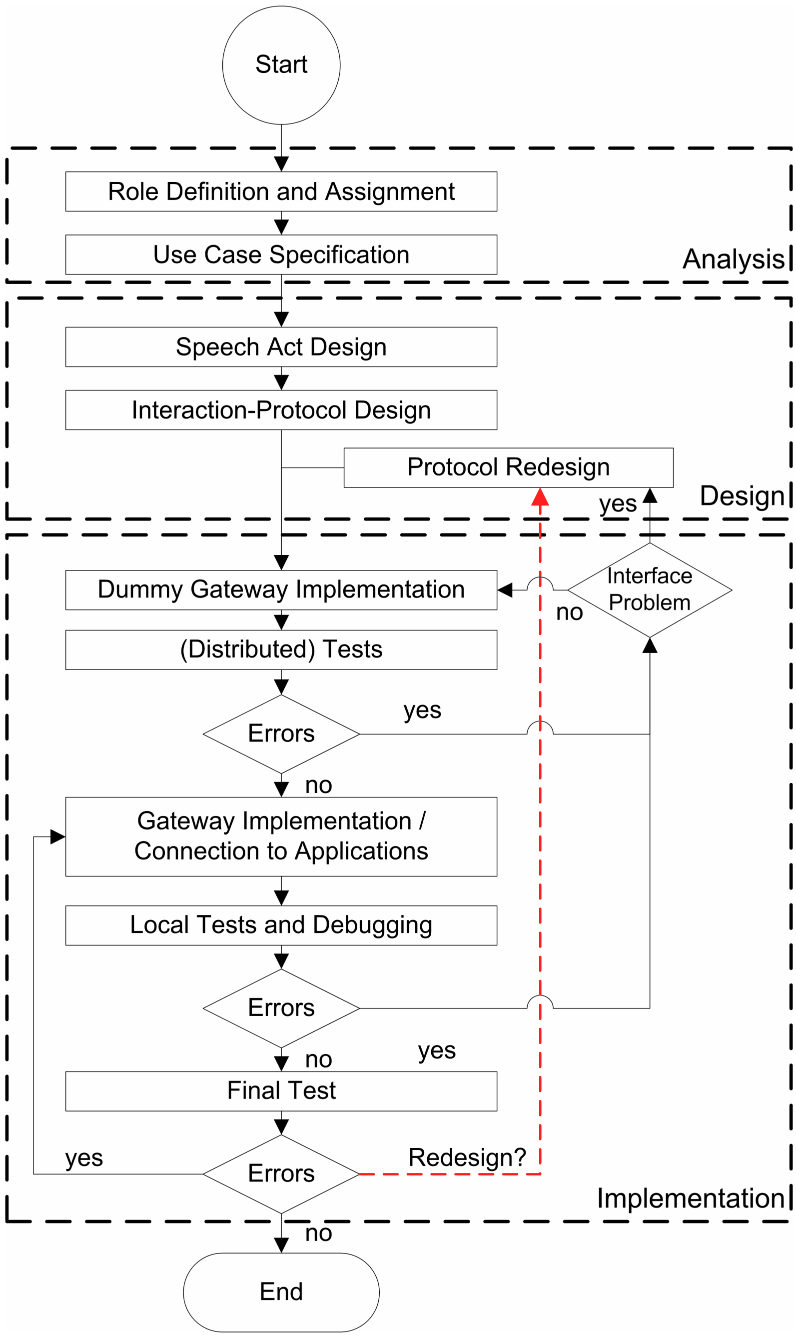

Fig. 2.1. The *Agent.Enterprise* development process

The task of ontological modeling is performed using the method described in [NM01], which was supported by an ontology modeling tool. There is a number of tools, which support ontology-modeling, ranging from Protégé [Gen02], OilEd [Bech01] to WebOnto [Do98], of which Protégé proved to be the best suited for the task of ontology modeling in a multi-agent environment. The Protégé-plug-in Beangenerator [Aart02] offers the functionality to generate code for the modeled ontology that is applicable with the JADE agent platform, which is used in several projects as development framework.

A starting point for the ontology modeling is to identify the actions of the agents, which are requested from their communication partner. They are directly derived from the assigned functionalities and specify which tasks are to be performed. At this point, the ontological concepts defining the artifacts to be dealt with in the agent actions can be specified, like e.g. the products to be manufactured. After modeling all the details for the supply chain, the concepts required for the supervision of all participating MAS are designed.

Interaction Protocol Design

The next step in the overall design process is to define the dynamics of the conversations, i.e. which interaction protocols will be used for communication. The informal specification of interactions resulting from the card role-play is mapped to corresponding FIPA interaction protocols. As a result, the behavior of each gateway-agent for each MAS is specified as far as communication between the gateways is concerned. The final step of the coupling process is to realize communication between the gateway-agent and the underlying MAS.

(Distributed) Implementation

Based on the distributed structure of our research program, the development process takes into account long periods of independent development. Inspired by the concepts of Extreme Programming [Beck99], the process starts with the implementation of functionally restricted prototypes executing a simplified test case. These prototypes serve two purposes. On the one hand a consolidation of speech acts and interaction protocols is enforced. On the other hand, a test bed for autonomous development emerges. The implementation of these prototypes is closely bound to test-sessions and frequently requires a redesign of the speech acts and/or interaction protocols. The outcome of this work is a set of test modules and an exchange of experience within the covering research program. A simplification of the central projects' functions in the 'Dummy Gateway Implementation Cycle' could prove to be a downside of our approach. This fact might result in a setting where some aspects of the interaction protocols could not be sufficiently tested.

Subsequent to the completion of the prototypes, each project integrates its fully functional application into the test bed. Sometimes a project has to debug prototypes of other projects. Explicit phases for consolidation are not necessary due to the strictly defined responsibilities for each prototype.

2.2 The Gateway-Agent Concept

Integration of complex systems requires agreements of architectural and technical nature in order to avoid a time-consuming struggle with implementation details. For *Agent.Enterprise* two central design decisions are subsumed in the so-called Gateway-Agent Concept. This is illustrated in Figure 2.2.

Firstly, the agreement upon the use of FIPA-compliant platforms avoids many of the communication-related obstacles and allows for concentrating on domain aspects. The second decision is that every individual MAS to be integrated should be represented by a single agent, that comprises all roles of its corresponding MAS, and provides them to the resulting MMAS.

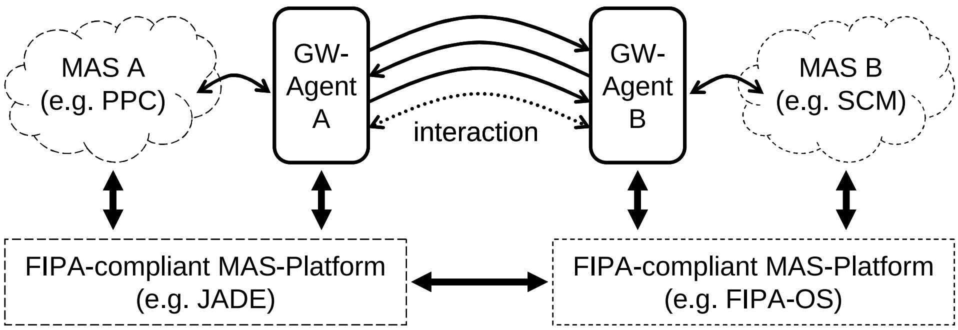

Fig. 2.2. The Gateway-Agent Concept

Thus, the interacting gateway-agents build up a virtual MAS. Together, these decisions can be seen as a specific MMAS architecture, which subsumes aspects of various well-known software patterns [Gam95].

- Façade pattern: The gateway-agents provide a unified interface to their MAS as a subsystem, comprising different roles and their respective functionality.
- Wrapper pattern (called Adaptor in [Gam95]): The gateway agents translate between internal formats and behavior of their corresponding MAS and the common representation in the virtual MAS.
- Bridge pattern: The different types of the gateway agents provide abstract interfaces decoupled from the implementation of their MAS. In the next section we give an example of three MAS that play the role of a supplier in a supply chain scenario. In the virtual MAS they all are represented by the same type of gateway agent while their implementation is completely different and independent.

There are many advantages of the Gateway-Agent Concept, e.g., only the gateway-agents must be available and externally visible for the other MAS during operation. In the earlier implementation phases developers can put their effort on the gateway-agent, when debugging the interoperability between the different MAS. Also, there is no restriction in the centralization of different roles of a MAS into a single gateway-agent. This is due to the fact that different functionalities of this agent can be redirected to several other agents in the MAS, which are represented by the gateway-agent.

3 The *Agent.Enterprise* System

The *Agent.Enterprise* System is the result of successfully applying the presented concept in the manufacturing domain. In the following subsection the MMAS is detailed to better understand the benefits of coupling MAS.

3.1 Scenario

The complexity of managing supply chains results in many different interdependent tasks such as planning, executing and controlling of production, transportation and warehousing processes. As a consequence, different MAS specializing on certain tasks have to interact with each other. The basic scenario focuses on production processes, whereas aspects of transportation can be integrated. Table 3.1 provides an overview of the various functionalities of the involved MAS.

Main Functionality	**Project/MAS**
Negotiations between enterprises	DISPOWEB
Integrated process planning and scheduling (with focus on discrete manufacturing)	IntaPS
Production planning and controlling (with focus on assembling industries)	KRASH
Production planning and controlling (with focus on batch production)	FABMAS
Operational tracking of orders including suborders in supply chains	ATT*
Analysis of historical tracking information (tracing)	SCC*

* ATT and SCC conducted in one project [ATT/SCC]

Table 3.1. Overview of individual MAS functionalities

A typical supply chain management cycle of distributed global planning as a part of the supply chain activities is shown in Figure 3.1. After generating an initial plan of orders and suborders comprising prices and points in time of delivery, software agents located at the different supply chain partners carry on negotiations. Thereby, they optimize the costs and the due delivery dates (see Figure 3.1, ①).

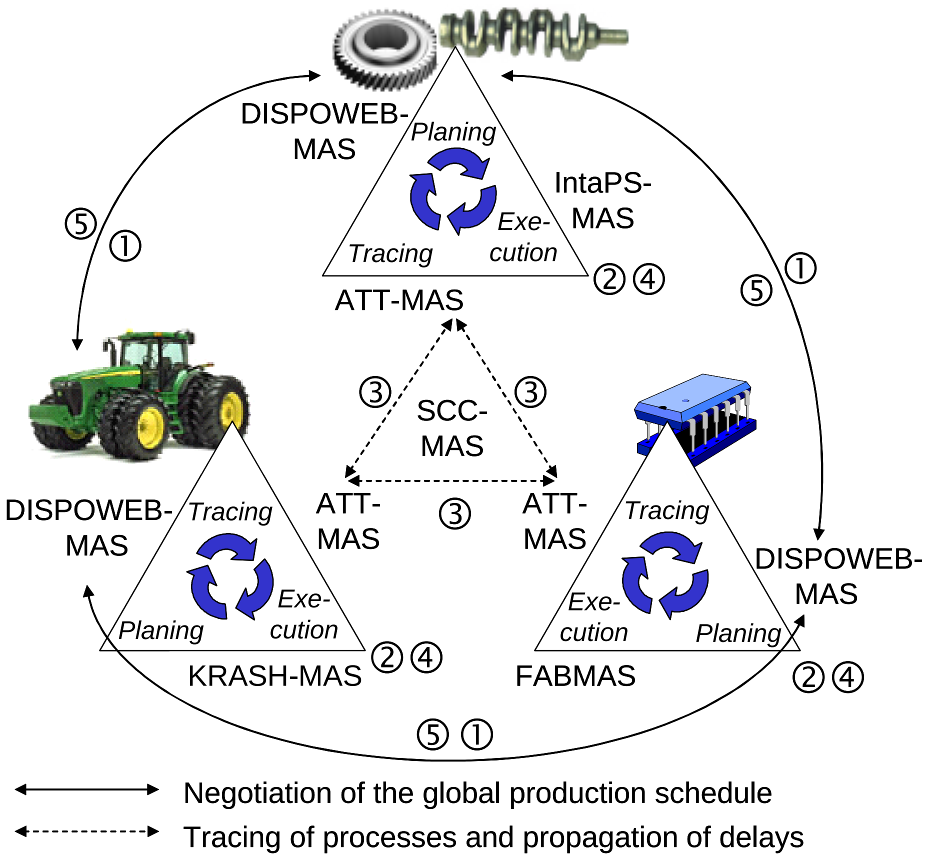

Fig. 3.1. MAS interaction in the integrated SCM architecture

These optimized delivery plans are used on the intra-organizational level to coordinate the production of goods on each stage of the supply chain in detail. Three different MAS [FABMAS] [KRASH] [IntaPS] are concerned with varying aspects of production planning (②). They require the input from DISPOWEB agents and generate detailed plans for their production facilities [DISPOWEB].

These plans are the initial input for a controlling system, which is developed in the ATT/SCC project. This MAS monitors orders on every stage of the supply chain using a distributed architecture in order to proactively detect events that endanger the planned fulfillment. In case of such an event, e.g. a disruption in a production cycle, the ATT system informs the related partner enterprises about the event (③). This information can be used to trigger the re-scheduling of plans on an enterprise level (④) or, in case of major events, even in the re-negotiation of contracts on the inter-enterprise level of the DISPOWEB system (⑤). An overview of activities and corresponding actors in the supply chain are given in Table 3.2.

Nr.	**Activity (Actor)**
①	Negotiate initial plan of production between supply chain partners (DISPOWEB).
②	Operational assembly planning (KRASH). Production planning for e.g. mechanical parts (IntaPS). Production planning for e.g. electronic parts (FABMAS).
③	Monitoring of orders and related suborders (ATT). Trigger internal planning systems in case of minor critical events (ATT). Next → ④ Trigger re-planning by DISPOWEB agents in case of a severe critical event (ATT). Next → ⑤ Controlling information is forwarded to trusted third party SCC-system (ATT/SCC).
④	Internal rescheduling in reaction to a critical event (KRASH, IntaPS, FABMAS). Next → ③
⑤	Renegotiate plan of production between supply chain partners due to severe critical event (DISPOWEB). Next → ②

Table 3.2. Activities and actors

In addition to the operational monitoring of orders, the ATT system communicates results of negotiations to a trusted-third-party service called the SCC-MAS. This agent system analyzes the history of orders and their related sub-orders. SCC is able to identify patterns in the supply chain and order types that typically lead to problems during fulfillment. This information is used as an input to enhance the tracking functionality of the ATT systems, as well as an input for the DISPOWEB agents to enhance their negotiation strategies, e.g. charging lower prices from suppliers with bad performance.

3.2 Benchmarking

Manufacturing systems have to provide flexibility and robustness to stay competitive. Multi-agent systems are expected to be more flexible than monolithic systems. In addition, special mechanisms integrated into the SCM reference model ensure the reliability of the MAS. Three features assert the flexibility and reliability of the supply chain.

1. Flexibility of the SCM is achieved by a distributed structure of optimization algorithms.
2. Proactive tracking and tracing methods are integrated into the reference model.
3. The distributed local shop floor PPC algorithms must be robust.

The first and the second feature have been discussed in the preceding subsection (3.1). The robustness of the shop floor planning systems is the basis for higher-level reliability of the SCM system.

On this intra-organizational level KRASH, IntaPS and FABMAS offer MAS-based PPC functionality. In the scope of the presented supply chain reference model, the shop floor MAS architectures match the requirements defined above. The statement that MAS are more flexible and robust than traditional planning systems can be verified using realistic benchmarking scenarios. The throughput times of productions orders were analyzed, whereas both the medium throughput time and its standard deviation were checked. The planning process was performed by a reactive MAS approach on one hand and a common OR algorithm on the other hand.

The evaluation shows, that the suitability of the PPC MAS depends on the frequency of disturbances. Nevertheless the standard deviation for the MAS was permanently lower throughout the whole evaluation process. Figure 3.2 shows a scaled comparison factor representing the probability of the MAS, respectively the OR algorithm to perform better. Values larger than 0 represent scenarios where the MAS is superior to the OR approach. It is obvious that this statement is valid for this example. The results even improve when the planning complexity of the scenario is increased (equals a decreasing lot size in this example). For further information concerning the OR algorithm and the evaluation process refer to [Frey03a].

Constant (or nearly constant with at least a small standard deviation) and thus predictable throughput times are a prerequisite for high quality results of the DISPOWEB planning MAS. Along with the ATT-MAS, this robustness on the operational level takes care of the overall robustness of the integrated SCM architecture.

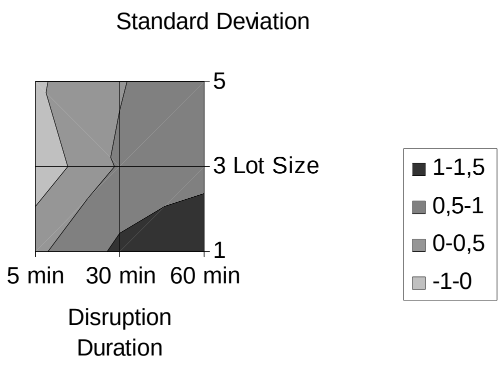

Figure 3.2. Suitability of a MAS approach with respect to the standard deviation of the throughput time

A potential disadvantage of the architecture is the increased communication and co-operation effort. This question has to be investigated in the next phase of the priority research program by testing the implemented MAS using realistic test case scenarios. One instantiation of a benchmark scenario is available at http://www.is-frankfurt.de/tractor. The chosen supply chain represents a tractor manufacturer and its suppliers. The transfer of the results to a broad spectrum of applications is possible due to the closeness of the test case to the automotive industry.

4 Conclusion

This paper presents our approach to couple MAS by combining and applying aspects from well-known agent-oriented development methods. Consequently, it accounts for the special needs that arise from the nature of MAS integration and the technical system architecture agreements. Moreover, its design fits the needs for distributed development.

The existence of sufficiently specified concepts does not require additional conceptual effort since these concepts are applicable for most steps in our proposed development process. Nevertheless, the introduced technical concept, called Gateway-Agents, accelerates the implementation by forcing necessary agreements on technical standards. The concept is presented in this paper as an integral part of the *Agent.Enterprise* development.

The resulting MMAS covers services in the range of supply chain scheduling, shop floor production planning and control, and proactive tracking and tracing services and represents a large-scale adaptable research prototype that guarantees the reliability of overall supply chain processes. The reference architecture and its interfaces and gateways are tested on the basis of a test case scenario. The goal of the evaluation is to prove the feasibility of the approach and gather first insights and results.

The objective of the participating projects in the *Agent.Enterprise* initiative is the enhancement of the existing work to a demonstration and evaluation platform, which supports the presentation and benchmarking of the subprojects.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) within the German priority research program 1083 "Intelligent Agents in Real-World Business Applications" (refer to http://www.realagents.org for further information).

References

[Agentcities] "Agentcities" project, [Internet], Available from: http://www.agentcities.org [Accessed 14.1.04]

[Aart02] van Aart, C. J.; Pels, R.F.; Giovanni, C.; Bergenti, F.: Creating and Using Ontologies in Agent Communication. *Proceedings of the Workshop on Ontologies in Agent Systems* held at the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2002). Bologna, Italy, 2002.

[ATT/SCC] "Agent-based Tracking and Tracing of Business Processes" project, [Internet], Available from: http://www.wi2.uni-erlangen.de/research/ATT/index-e.html [Accessed 14.1.04]

[Bech01] Bechhofer, S.; Horrocks, I.; Goble, C.; Stevens, R.: OilEd: A Reason-able Ontology Editor for the Semantic Web. *Proceedings of Joint German/Austrian conference on Artificial Intelligence (KI2001)*. Springer-Verlag, Heidelberg, Germany, 2001.

[Beck99] Beck, K.: *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Reading, Mass., 1999.

[BPR01] Bellifemine, F.; Poggi, A.; Rimassa, G.: JADE: A FIPA2000 Compliant Agent Development Environment. *Proceedings of the Fifth International Conference on Autonomous Agents (Agents01).* ACM Press, 2001.

[CP02] Cosentino, M.; Potts, C.: A CASE Tool Supported Methodology for the Design of Multi-Agent Systems. *Proceedings of the International Conference on Software Engineering Research and Practice*. CSREA Press, 2002.

[DISPOWEB] "Dispositive Supply-Web-Coordination" project, [Internet], Available from: http://www.dispoweb.de [Accessed 14.1.04]

[Do98] Domingue, J.: Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. *Proceedings of the 11th Workshop on Knowledge Acquisition for Knowledge-based Systems*. Banff, Canada, 1998.

[FABMAS] "Agent-Based System for Production Control of Semiconductor Manufacturing Processes" project, [Internet], Available from: http://www.tu-ilmenau.de/fabmas [Accessed 14.1.04]

[FIPA] "Foundation for Intelligent Physical Agents" standardization organization, [Internet], Available from: http://www.fipa.org [Accessed 14.1.04]

[Frey03a] Frey, D.; Nimis, J.; Wörn, H.; Lockemann, P.: Benchmarking and Robust Multi-agent-based Production Planning and Control. *Engineering Applications of Artificial Intelligence - The International Journal of Intelligent Real-Time Automation (EAAI)*, Volume 16, Issue 4 , Pages 307-320, 2003.

[Frey03b] Frey, D.; Stockheim, T.; Woelk, P.-O.; Zimmermann, R.: Integrated Multi-agent-based Supply Chain Management, *Proceedings of the 1st International Workshop on Agent-based Computing for Enterprise Collaboration*. IEEE Computer Society Press, 2003.

[Gam95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, Reading, Mass., 1995.

[Gen02] Gennari, J.; Musen, M. A.; Fergerson, R. W.; Grosso, W. E.; Crubézy, M.; Eriksson, H.; Noy, N. F.; Tu, S. W.: The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. *Medical Informatics Tech. Report SMI-2002-0943*. Stanford, 2002.

[IntaPS] "Integrated Agent-based Process Planning and Production Control" project, [Internet], Available from: http://www.intaps.org [Accessed 13.3.03]

[KRASH] "Karlsruhe Robust Agent Shell" project, [Internet], Available from: http://www.ipd.uka.de/KRASH/ [Accessed 14.1.04]

[Lind01] Lind, J.: Iterative Software Engineering for Multiagent Systems: the MASSIVE Method. *Lecture Notes in Computer Science and Lecture Notes in Artificial Intelligence*, Vol. 1994, Springer-Verlag, 2001.

[NM01] Noy, N. F.; McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Tech. Report SMI-2001-0880*. Stanford, 2001.

[OVB01] Odell, J.; Van Dyke Parunak, H.; Bauer, B.: Representing Agent Interaction Protocols in UML. *Proceedings of the 1st International Workshop on Agent-oriented Software Engineering.* Springer-Verlag, 2001.

[Sto04] Stockheim, T.; Nimis, J.; Scholz, Th.; Stehli, M.: How to Build Multi-Multi-Agent Systems: the Agent.Enterprise Approach. *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS).* Porto, Portugal, 2004.

[WD01] Wood, M.F.; DeLoach, S.A.: An Overview of the Multiagent System Engineering Methodology. *Proceedings of the First International Workshop on Agent-oriented Software Engineering*. Springer-Verlag, 2001.

[WJK00] Wooldridge, M.; Jennings, N. R.; Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 3, 2000, pp. 285–312.