AN AGENT-BASED COOPERATION PLATFORM FOR FIRE PROTECTION PLANNING

Uwe Rüppel[1] , Udo F. Meißner[2], Michael Lange[3]

ABSTRACT

One major aim in computer-aided engineering is to support the engineers in communication and collaboration during the building planning process. In order to realize this, new approaches and methods are being developed in the priority research program 1103 "Network- based cooperative planning processes in civil engineering" of the Deutsche Forschungsgemeinschaft. The presented project is part of this priority research program.

Since the model based planning becomes accepted in the building industry we assume planners to work on partial models. As the planning process of building projects is characterized by the cooperation of many experts the partial models are highly interrelated. This paper presents an agent-based cooperation platform to support planners involved in fire protection planning by linking the distributed partial models together, preserving their consistency and giving the possibility of an automated regulations-check-up of the models. Different types of agents and their roles in the cooperative process are described. Agents for communication, for the acquisition of information and for knowledge processing in fire protection engineering are introduced. The possibility to check-up planning information for consistency and compliance with the fire protection regulations at an early state of the planning process enables a comprehensive diagnosis of the design and leads to a reduction of planning errors.

KEY WORDS

Cooperation, distributed planning processes, software agents.

INTRODUCTION

In this contribution a cooperation platform to manage distributed planning processes on the example of fire protection planning is presented. By the help of (software-) agents distributed technical information of each involved planner is integrated into the planning processes. The vision is to provide agents as planners digital representatives moving through computer networks during the building design phase and in doing so, monitoring relevant planning activities, searching for distributed planning information, checking-up the building design - here in terms of fire protection planning - and communicating or negotiating with other

1 Professor, Institute of Numerical Methods and Informatics in Civil Engineering, Department of Civil Engineering and Geodesy, Darmstadt University of Technology, Petersenstrasse 13, Darmstadt, D-64287, PH (+49) 6151-163444, FAX (+49) 6151-165552, sekretariat@iib.tu-darmstadt.de
2 Professor, contact: see above
3 Research Assistant, contact: see above

agents. Such an agent is a mobile and autonomous software component supporting fire protection engineers in their everyday engineering tasks.

The presented project is part of the priority research program 1103 of the German Research Foundation (DFG; www.dfg-spp1103.de). This contribution covers the main research results of the project "Agentenverbund für die kooperative Gebäudeplanung" that has been part of the priority program 1103 over 6 years.

DFG SPP 1103

Considering powerful computer networks and the intensive globalization in science and technology in the field of civil engineering, the priority research program "Vernetzt-kooperative Planungsprozesse im Konstruktiven Ingenieurbau" aims at redesigning the planning processes of civil engineering for the utilization of distributed resources. Further objectives are developing adequate cooperation models for the technical planning by use of information sharing between project partners and allowing cooperative project work with distributed technical models in networks. In this context, new methodical basics are to be developed to consider the profound changes of industrial and organizational structures. The complexity of the problem area requires an intensive interdisciplinary cooperation of the involved research disciplines focusing on the field of Informatics in Civil Engineering ("Bauinformatik"). The whole priority program combines 14 projects with nearly 30 researchers and has a duration from 2000 to 2006. After 2006, research transfer projects are planned to implement the research results into applied products. This research program is coordinated by the authors.

AN INTEGRATIVE PROCESS MODEL

In order to adequately support the cooperation of planning participants in heterogeneous computer networks, appropriate software methods have to be developed. These software methods have to rely on a comprehensive process model. In (Meißner et al., 2002) the idea of an integrative process model comprising of four layers is developed:

- The resource layer storing objects and their model states during the planning process. Furthermore, this layer includes the rules and methods needed to process the model information.
- The coordination layer representing the flow of work within planning. Petri-Nets (Petri, 1962) supply a suitable theory to model these processes (Katzenbach et al., 2002).
- The actor layer modeling planners and organizations involved in the process and having control over models and decision making.
- The communication layer modeling the dynamic interaction flow of information between the planning participants. This layer is the most important for direct access to information, based on modern communication technologies like mobile software-agents (Rüppel et al., 2002a).

Figure 1 illustrates the approach of an integrative process model for Civil Engineering and its four layers.

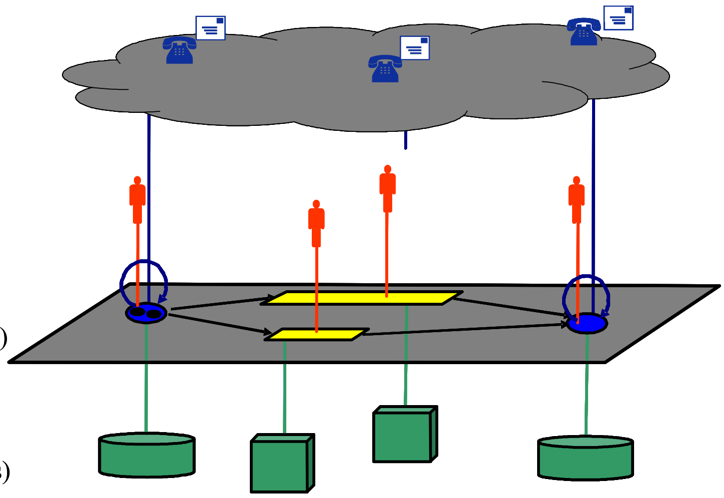

Figure 1. Integrative Process Model for Civil Engineering.

MODEL-BASED FIRE PROTECTION PLANNING

In building design, fire protection planning is a dominant aspect for the protection of life and property in case of fire. Fire protection engineering is divided into two domains: the preventive and the defensive fire protection. The preventive fire protection contains all structural, technological and organizational fire protection aspects. The fire fighting and rescue are aspects of the defensive fire protection. Within the preliminary planning the elementary requirements for an effective personal safety and an optimal fire-fighting are created by the preventive fire protection (Schneider & Lebeda, 2000). For that purpose, the building geometry and the adjustment of escape routes are important.

According to the type and size of a project, planners from different fields (e.g. statics, construction, heating and ventilation) are involved. Methods to validate the planning results with regard to completeness and effectiveness concerning the fire protection objectives must be provided. Therefore, a large number of codes and regulations have to be evaluated. The preventive fire protection planning has to be considered during the whole planning and construction phase and is focused in the presented approach.

For every building a fire protection concept has to be created to describe all planned methods to prevent fire. In order to process a fire protection concept in a distributed network of planners a transparent model is necessary. Thus, in (Rüppel, et al., 2002a) a new fire protection model has been developed and implemented in the CAD-System Autodesk Architectural Desktop (figure 3).

During creation of the fire protection model, the new information is directly associated with the building model. A corridor in the building model, for example, can be identified explicitly as an escape route. The definition of an escape route causes special demands on the linked building components. These requirements must be permanently checked during the planning process (Rüppel et al., 2002b).

MODEL INTEGRATION IN AN AGENT-BASED COOPERATION PLATFORM

The presented approach for cooperation support is based on the assumption that every planner works on a partial model, for example, the fire protection model, the structural model or the building services model. In distributed planning environments, the integration of partial models stored in databases is an essential task. In the presented work, an approach based on software agents is developed to support the model integration (Rüppel et al., 2002a).

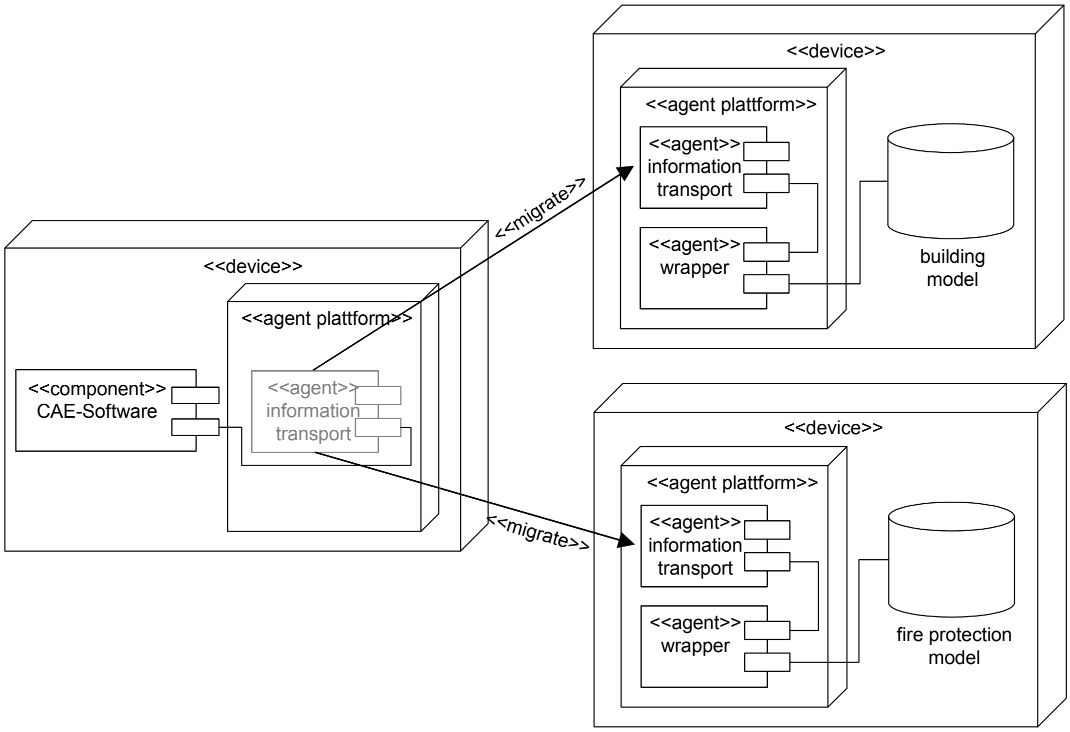

Figure 2. Agent-based model integration.

A software agent is a software module which operates in a special software environment, the agent platform. The environment sets up on the operating systems and establishes connections to other software applications or databases. The essential difference between objects - in terms of object-oriented programming - and agents is the achievement of objectives. Objects are defined by their attributes and methods. An object encapsulates the states of the attributes and contains a variety of methods which are used to change the states. Due to their inner state and their environment, agents decide whether actions are executed. An agent can also refuse to fulfill orders due to its knowledge (Bigus & Bigus, 2001). Therefore, an agent contains the knowledge and objectives to fulfill its intentions.

To realize the integration of partial building models every model has a proxy in form of a stationary wrapper agent on its platform (Theiß & Lange, 2003). This agent answers questions regarding the underlying model representing the planning details. The transportation of the model information is supported by mobile information agents (figure 2). This cooperation platform architecture supports a flexible integration of the partial models and is specified in the following chapters.

RESOURCE WRAPPING BY AGENTS

Database wrapper agents have been developed to integrate design information into the multi-agent system. They offer the relevant product model data stored in heterogeneous databases to other agents independent of their physical location. Thus, database wrapper agents act as an interface between the multi-agent system and heterogeneous database systems (Bilek & Hartmann, 2003). The communication between the database wrapper agents and other requesting agents implies a common vocabulary. A specific database ontology maps database related message contents to database objects. The software-wrapping technology thereby enables the various design experts to plug existing database systems and data resources into a specific multi-agent system. Consequently, dynamic changes in the design information of large collaborative engineering projects are adequately supported (Hartmann et al., 2004).

The Foundation for Intelligent Physical Agents (FIPA) has incorporated the concept of wrapper agents in its specifications for the design of multi-agent systems. Wrapper agents are usually implemented in terms of stationary agents. They act on the host where the applied resource is located. Wrapper agents provide an easy to use interface to the services provided by the integrated resources. Thus, they hide the internal database structure required for accessing the encapsulated data resources. The architecture of the database wrapper agent is introduced in (Theiß & Lange, 2003).

COMMUNICATION IN AN AGENT-BASED COOPERATION PLATFORM

Besides the wrapping of the databases storing the partial building models, the communication among agents in the cooperation platform has to be defined to achieve flexible model integration. Three specific problems have to be considered: firstly, the information transport in the multi-agent system, secondly, the structure of the communication process and thirdly, the syntax and semantics of the message content (Rüppel et al., 2005a).

Information Transport

Different partial product models have to be retrieved by agents for processing purposes. The transportation of the model information is supported by mobile information agents. The information agent presented in (Meißner et al., 2004) offers the service of information transport in the network-based planning environment. The information transport is divided into two steps: the transport of the query to the database wrapper agent and the transport of the response from the database to the initiator of the query.

The multi-agent system consists of several agent-platforms. One platform can comprise several computers, whereas an agent can migrate within this platform. For the inter-platform migration of agents a new service has been developed (Meißner et al., 2004). Every platform with inter-platform agent migration support has to instantiate this service. The service offers the sending and receiving of agents. The process of agent migration contains several steps: after starting the migration process, the agent classes and instances are packed. Then the result is transformed into the content of an Agent-Communication-Language (ACL) message and hashed by a MD5 hash key for security reasons. Finally, the message is sent to the agent migration service of the destination platform. The process of reactivating the agent after migration corresponds to the sending process in reverse order. Thus, a multi-agent system with inter-platform-mobility could be realized.

Structure Of Communication

After migrating through the network, the information agent gets in contact with a stationary agent by sending Agent-Communication-Language (ACL) messages. The internal structure of an ACL Message is defined by the FIPA (FIPA 2003). To structure the exchange of messages, the FIPA has defined so-called agent interaction protocols (IP) (FIPA, 2002b) and communicative acts (CA) (FIPA, 2002a). A CA describes the object of an ACL message, for example 'query', 'inform' or 'subscribe'. An IP describes a complete communication dialogue between two interacting agents. To model the dialogue between information and wrapper agents a so-called information-request-interaction-protocol is implemented in the described agent system (Theiß, 2005).

Semantics Of Communication

In addition to the communication acts and realized interaction protocols, ontologies have been developed which define the homogeneous vocabulary used in the message contents. A request ontology determines the query model defining the three action tags *"select"*, *"insert"* and *"update"*. The request ontology is fundamental for each database wrapper agent implementation because it matches the agent's core functionality. Depending on the project related tasks and knowledge, ontologies are necessary to describe the specific technical contents and product models. In the presented project, an ontology defining the building structure and the corresponding fire protection elements has been implemented (Hartmann et al., 2004). Furthermore, the mapping of the entities of the public ontology to the entities of the proprietary database schema has to be defined. Thus, queries on the basis of the ontology can be translated into proprietary database queries and the result can be translated into instances on the basis of the public ontology and then passed to the information agent.

INFORMATION PROCESSING WITH AGENTS

As mentioned above in fire protection engineering a large number of codes and regulations have to be evaluated. Thus, on the basis of the described agent-based model integration a automated check-up of the building models for compliance with the fire protection regulations has been developed.

The regulations in fire protection are declaratively styled. Rule-based expert systems are well known in processing declarative rules. The processing of fire protection rules in an expert system to check-up the compliance of the building model with the fire protection guidelines has been realized by the efficient combination of model data and rules. Therefore, a three-tier knowledge model for preventive fire protection has been developed: firstly, it consists of the data models for the building; secondly, a rule model has been developed to represent the guidelines and regulations of fire protection; and thirdly, it consists of the fire protection model for the building.

For the application of the fire protection rules to the data models, the rules must be defined and structured in a processable form. Therefore, the rules are structured in a first step by the type of the regulation and in a second step by the related building element. The rule structure optimizes the validation process as well as the communication process for acquiring necessary rules (Meißner et al., 2003). By means of a problem-specific user interface implemented on the basis of the ontology editor Protégé (Protégé, 2005) the CLIPS-based rules (Riley, 2005) can be graphically defined, so that they can be processed and integrated in the structure described above (Theiß, 2002).

To realize the processing of the rules a "fire protection agent" has been presented in (Rüppel et al., 2002a). This agent is able to process the facts of the building's fire protection model with the requirements from the design codes by the integrated expert system Jess. Jess (Java Expert System Shell) (Friedmann-Hill, 2003) is one of the most used rule-based expert systems and is the reference implementation for the Java Rules Engine API. Rule-based systems process rules and facts in an inference engine. An example for a fact is the width of an escape route; the corresponding rule is that every escape route must have a minimum width of 1.25 meters.

PETRI-NET BASED AGENT CONTROL

Previously, agents to integrate and process planning information in the cooperation platform to support planers in there everyday engineering tasks concerning fire protection engineering have been presented. Thus, the resource layer of the integrative process model (described above) providing information objects, rules and methods can be adequately supported.
To structure and organize the engineering tasks (coordination layer of the integrative process model (figure 1)) an appropriate modeling method is necessary. Petri-Nets supply a theory for the modeling and control of concurrent and asynchronous processes with strong semantics.

The combination of agent oriented methods and the Petri-Net method is an ongoing research topic in computer science. The presented approach is based on two net types (Rüppel et al., 2003). The first net type contains the planning activities, planning states, planning information and the sequential, iterative or concurrent dependencies. This net is denoted as the "planning net". Within this net the transitions represent planning activities or events, respectively. An agent executing a specific planning activity can be associated with a transition. However, the agent itself is not an element of this net. The second net type describes the mobility of an agent which supports a special planning task. Hereby, a transition models the agent's migration and the agent's interaction with other agents during runtime. As the focus lies on the agent modeling, this net is referred to as the "agent net". The combination of these two net types, namely planning net and agent net, enables a holistic modeling of the agent-based support of planning processes with a unique semantics.

EXAMPLE OF USE

In the following a short example of a fire protection requirement check-up is modeled. In this context a combination of the presented methods will be shown in the planning scenario: firstly, wrapper and information agents to integrate the distributed planning information, secondly, the fire protection agent that provides knowledge-based methods to process this information and thirdly, Petri-Nets to coordinate the workflow of the planners and corresponding agents.

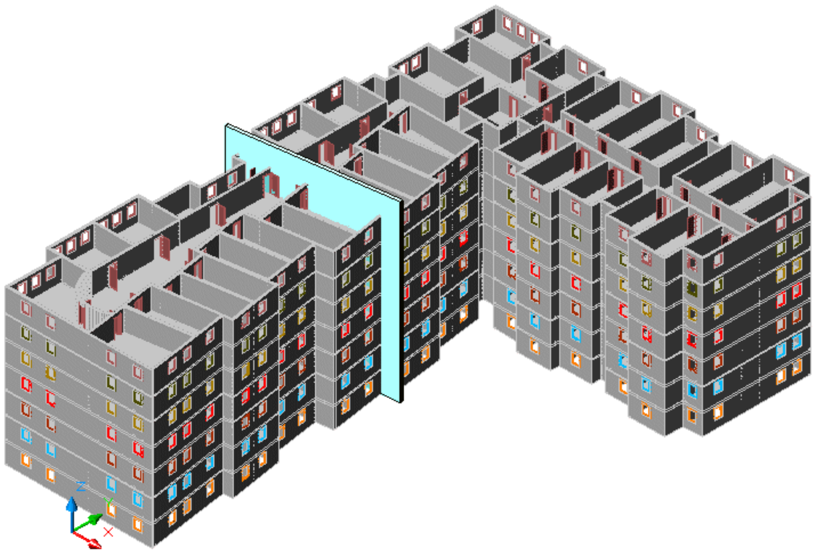

Figure 3. Building model with a fire break wall separating two fire compartments.

To avoid the spread of fire, fire break walls have to be used to separate fire compartments. In this example two fire compartments are planned in a floor of an existing building. This design task is based on a close communication and collaboration between the architect, the structural engineer and the fire protection engineer.

After adding the two fire compartments the fire protection agent is assigned to check the building elements for compliance with fire protection regulations. The fire protection agent commissions an information agent to procure the relevant design information. The information agent migrates through the network, communicates with different stationary wrapper agents, exploits the services of a directory facilitator agent, requests information from databases and finally carries the information back to his source location, i.e. the workstation of the fire protection agent.

Thereupon, the fire protection agent checks-up the design information for compliance with the fire protection rules by the built-in expert system and gives feedback to the fire protection engineer and the structural engineer: the wall between the fire protection compartments has to be a fire break wall (figure 3) and hence has to meet some special fire protection and structural requirements. The flow of work of all involved agents and engineers is modeled as “planning and agent nets”.

CONCLUSIONS

In order to adequately support the cooperation of planning participants in heterogeneous computer networks, appropriate software methods have been developed in the presented project. After introducing the integrative process model to organize and support distributed engineering processes in computer networks this contribution presents an agent based cooperation platform to realize the coordination and resource layer of this model.

The presented prototype supports fire protection engineering by integrating partial building models in the distributed planning processes. Furthermore, it enables the processing of this information to check-up the compliance of the distributed partial building models with fire protection regulations. To structure and organize the agent supported engineering tasks Petri-Nets for the modeling and control of concurrent and asynchronous processes are presented. Finally, the design of a fire break wall as a typical task in fire protection engineering was shown as an example. The presented result enables a high quality and efficient cooperative project work with distributed technical models in networks.

REFERENCES

Bigus, J. & Bigus, J. (2001). Intelligente Agenten mit Java programmieren. Addison-Wesley, Munich, Germany.

Bilek, J. & Hartmann, D. (2003). Development of an Agent-based Workbench supporting Collaborative Structural Design. In: The 20th CIB W78 Conference on Information Technology in Construction. New Zealand.

FIPA (2002a). FIPA Communicative Act Library Specification (available at http://www.fipa.org/specs/fipa00037/index.html).

FIPA (2002b). FIPA Interaction Protocols (IPs) specifications (available at http://www.fipa.org/repository/ips.php3).

FIPA (2003). ACL Message Structure Specification (available at http://www.fipa.org/specs/fipa00061/index.html).

Friedmann-Hill, E. (2003). Jess in action - Java rule-based systems. Greenwich: Manning Publications.

Hartmann, D.; Meißner, U. F.; Rüppel, U.; Bilek, J.; Theiß, M. (2004). Integration of Product Model Databases into Multi-Agent Systems. In: Proceedings of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X).

Katzenbach, R.; Meißner, U. F.; Rüppel, U.; Giere, J.; Greb, S. (2002). Process-oriented Network-based Collaboration in Geotechnical Engineering. In Proceedings of the 9th International Conference on Computing in Civil and Building Engineering Vol. II, Taiwan, 1063-1068.

Meißner, U. F.; Rüppel, U.; Theiß, M. (2002). Verteilte Brandschutzmodellierung auf der Basis von Software-Agenten. In: Bauen mit Computern - Kooperation in IT-Netzwerken, Bonn, Fortschritt-Bericht VDI, 557-569.

Meißner, U. F.; Rüppel, U.; Greb, S.; Theiß, M. (2003). Network-based Cooperation Processes for Fire Protection Planning. In: R. Amor (Ed.): Proceedings of the CIB W78's 20th International Conference of Information Technology For Construction. University of Auckland. New Zealand. 231-238.

Meißner, U. F.; Rüppel, U.; Theiß, M. (2004). Network-Based Fire Engineering Supported by Agents In: Proceedings of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X).

Petri, C. A. (1962). Kommunikation mit Automaten. Schriften des Instituts für Instrumentelle Mathematik der Universität Bonn. Germany.

Protégé (2005). Ontology editor and knowledge-based framework. Stanford Medical Informatics at the Stanford University School of Medicine (available at http://protege.stanford.edu).

Riley, G. (2005): C Language Integrated Production System (CLIPS) - A Tool for building Expert Systems (available at http://www.ghg.net/clips/CLIPS.html).

Rüppel, U.; Meißner, U. F.; Theiß, M. (2002a). Fire Protection Concepts With Mobile Agents. In: Proceedings of the 9th International Workshop of the European Group for Intelligent Computing in Engineering (EG-ICE). Darmstadt.

Rüppel, U.; Meißner, U. F.; Theiß, M. (2002b). An Agent-based Platform for Collaborative Building Engineering, In: Proceedings of the 9th International Conference on Computing in Civil & Building Engineering, Taipei, Taiwan.

Rüppel, U.; Greb, S.; Theiß, M. (2003): Managing Distributed Planning Processes in Fire Protection Engineering based on Agent Technologies and Petri-Nets. In: 10th ISPE International Conference on Concurrent Engineering: Research and Applications, Band 2, S. 651-656, Madeira Island, Portugal, Juli 2003.

Rüppel U.; Theiß, M.; Lange, M. (2005a): Agent-enabled Model Integration in a Knowledge-based Planning Environment. Proceedings of the 22nd Conference on Information Technology in Civil Engineering (cibW78), P. 471– 476, Dresden 2005 (ISBN 3-86005-478-3; CIB Publication No. 304).

Schneider, U. & Lebeda, C. (2000). Baulicher Brandschutz. Stuttgart: Verlag W. Kohlhammer.

Theiß, M. (2002). Dynamische Integration von technischen Wissen in den Bauplanungsprozess. In: Bilek: 14. Forum Bauinformatik, Fortschritt-Bericht VDI, Bonn, Germany.

Theiß, M.; Lange, M. (2003). Integration von XML-basierten Fachinformationen in Multiagentensysteme. In: Kaapke, Kai; Wulf, Alexander (Ed.): Forum Bauinformatik 2003 - Junge Wissenschaftler forschen. Hannover: Shaker Verlag.

Theiß M. (2005) Agentenbasierter Modellverbund am Beispiel des baulichen Brandschutzes in der Gebäudeplanung, Berichte des Instituts für Numerische Methoden und Informatik im Bauwesen Nr. 03/2005, TU Darmstadt, 176 S., 113 Abb., Shaker Verlag, Aachen, ISBN 3-8322-4359-3, 2005.