# AN INTEGRATIVE PROCESS MODEL FOR COOPERATION SUPPORT IN CIVIL ENGINEERING

SUBMITTED: December 2005
REVISED: May 2006
PUBLISHED: July 2006 at http://www.itcon.org/2006/37/
EDITOR: P. Katranuschkov

***Uwe Rueppel, Prof. Dr.-Ing.***
***Institute of Numerical Methods and Informatics in Civil Engineering, Department of Civil Engineering and Geodesy, Darmstadt University of Technology, Germany***
***email: sekretariat@iib.tu-darmstadt.de, www.iib.bauing.tu-darmstadt.de***

***Michael Lange, Dipl.-Ing.***
***Institute of Numerical Methods and Informatics in Civil Engineering, Department of Civil Engineering and Geodesy, Darmstadt University of Technology, Germany***
***email: lange@iib.tu-darmstadt.de, www.iib.bauing.tu-darmstadt.de***

***SUMMARY:*** *Planning processes in civil engineering are highly complex and characterised by a great number of participants and heterogeneous, distributed planning resources in terms of software methods, models and means of communication. The interaction, communication and information exchange between planning participants must be organised in computer networks. This paper presents an integrative process model to support cooperation in distributed planning processes in civil engineering, taking fire protection planning as an example. Firstly, the integrative process model consists of Petri-Nets to model the processes and to support the coordination between the participants during the planning process. Secondly, it consist of agents providing resources und methods to process the technical information within the modelled processes. Thirdly, Petri-Nets are used to model the agent migration and the agent interaction.*

***KEYWORDS:*** *Cooperation, distributed planning processes, process modelling, software agents, Petri-Nets.*

## 1. INTRODUCTION

### 1.1 Process-Orientation in Civil Engineering

The design process in civil engineering is characterised by a very complex interactivity, interdependence and communication between different engineers. The close cooperation of engineers is the basis of a high product quality, short development periods and a minimum of investment costs. The organisations of the planning teams vary dynamically, depending on the project and the state of the planning process.

Planning processes in civil engineering differ by specific characteristics from planning and design processes in other industry domains. Generally, processes in civil engineering are highly complex and characterised by the design of unique copies. A high number of planning participants is involved, whereby each planning participant is a specialist in his technical domain and has his own point of view on the building project and thus specific objectives. The planning resources are heterogeneous and distributed, concerning the software methods, models and means of communication.

Thus, the interaction, communication and information exchange between different planning partners must be organised in computer networks. To support the coordination of highly iterative and concurrent processes in civil engineering adequate collaboration methods are to be developed.

In other industry domains, like the manufacturing industry, the process-oriented approach is applied in order to improve business processes. This is – more or less – neglected in civil engineering until now. In contrast to model-oriented or application-oriented approaches, process-orientation does not zoom in on details of a single technical planning task, but it comprises multiple planning tasks and their temporal and logical dependencies. Process orientation stands for (Hollingsworth, 1995):

- identification of processes, i.e., to recognise and to name the real world planning and construction processes and their dependencies, either within a single enterprise organisation or even beyond the boundaries of multiple organisations,
- modelling of processes, i.e., to represent the real world planning and construction processes in appropriate and computer processible process models; formal process models are essential in order to guarantee well-defined syntax and semantics of consistent process models,
- analysis of processes, i.e., to verify the process models in order to avoid modelling mistakes, like lack of synchronisation or deadlock, or to optimise the process models in terms of quality, time, costs, resources, etc. and finally,
- management of processes, i.e., to adapt the processes to modifications due to new requirements or exceptions in the real world processes at all times.

General goals of process modelling are defined, e.g., in (Rump, 1999). They comprise the documentation, simulation and optimisation of processes and the certification according to ISO 9000 or the implementation of workflow- and information-systems. Crucial objectives of process modelling in civil engineering are:

**Improvement of the transparency of the planning processes:** Planning processes are highly complex. Therefore, it is quite difficult for an individual planning participant to overlook the impact of his planning decisions on other planning activities. The transparency of the planning processes can be improved for each individual participant by using appropriate process models. These models should comprise planning states, activities, resources and their dependencies.

**Coordination of the distributed planning processes:** Planning processes are characterised by a great division of work where independent organisations have to work together. These cooperative planning activities have to be coordinated to fulfil a common objective. Process modelling and appropriate software methods are essential for the coordination of distributed planning activities.

**Modelling of engineering knowledge**: During the planning process a lot of information is generated, processed and modified. The necessary technical knowledge is encoded in technical standards, literature, software tools and especially in the heads of each individual planning participant. Formal process models are, among other things, an appropriate method to document the engineering knowledge.

**Process-oriented information structuring**: During the planning process various documents and models are generated and stored either on central servers or distributed work stations. For the distributed planners it becomes more and more difficult to have the relevant information at hand at a distinct step in the planning process. In addition to document-based or model-oriented structuring methods, appropriate process models can support the structuring and retrieval of planning information.

In the last years, different process-oriented approaches have been developed, for example in (Karhu, 2001), (Huhnt, 2004) and (König et al. 2004). This paper presents a new and different comprehensive approach to process modelling by, firstly, focussing on the integration of model-based planning information by agents; secondly, the integration of technical knowledge – representing the technical guidelines – in the process oriented approach and thirdly, the workflow coordination of the involved human beings and their corresponding agents by process models.

## 1.2 Research Context

The support of network-based concurrent engineering processes by agent technology and process modelling, based on Petri-Nets, are current topics of the priority research program 1103 of the German Research Foundation (DFG, www.dfg-spp1103.de). Considering powerful computer networks and the intensive globalisation in science and technology in the field of civil engineering, the priority research programme 1103 aims at redesigning the planning processes of civil engineering for the utilisation of distributed resources. The goal is to develop adequate cooperation models for technical planning in order to allow cooperative project work with distributed technical models in computer networks. The improvement of quality and efficiency for distributed design engineering and the development of innovative methods of informatics in civil engineering, are of central interest. The whole priority programme combines 14 projects with nearly 30 researchers and lasts from 2000 to 2006. After 2006 research transfer projects are planned to implement the research results into practical products. This research programme is coordinated by the authors.

## 2. APPROACH FOR AN INTEGRATIVE PROCESS MODEL ON EXAMPLE OF FIRE PROTECTION PLANNING

### 2.1 Overview

As a result of huge fire disasters fire protection planning is one of the central aspects for administrative authorities in the process of licensing the building design. Optimal conditions for the rescue of persons in case of fire can only be established if the fire protection requirements are defined early in the planning process. The consistent realisation of the fire protection model in all details has high demands on the communication and collaboration of the involved engineers and the corresponding building models. To preserve the related design models consistent to each other and compliant with the regulations of fire protection planning is a complex task.

The paper presents an approach to manage distributed planning processes in terms of cooperation support, taking fire protection planning as an example. The centre of the approach is an integrative process model. With the help of (software-) agents the distributed technical information of each involved planner is integrated into the central process model. The vision is that agents as digital representatives of the planners move through the computer network during the building design phase. Furthermore, the agents should monitor relevant planning activities, search for distributed planning information, check the building design – here in terms of fire protection planning – and communicate or negotiate with other agents. Such an agent is a mobile and autonomous software component supporting the fire protection engineer in his everyday engineering tasks. To model the complex planning processes, an appropriate modelling method is necessary. Petri-Nets offer a theory for modelling, analysing and controlling of concurrent and asynchronous processes with strong semantics. Based on the process model, the agent support and an integrated term network representing technical guidelines, a knowledge-based planning environment for cooperation support is developed. The application of agent technologies and Petri-Nets is illustrated by a typical planning scenario in fire protection planning.

### 2.2 Integrative Process Model

In (Meißner et al., 2002) the idea of an integrative process model is developed (Fig 1).

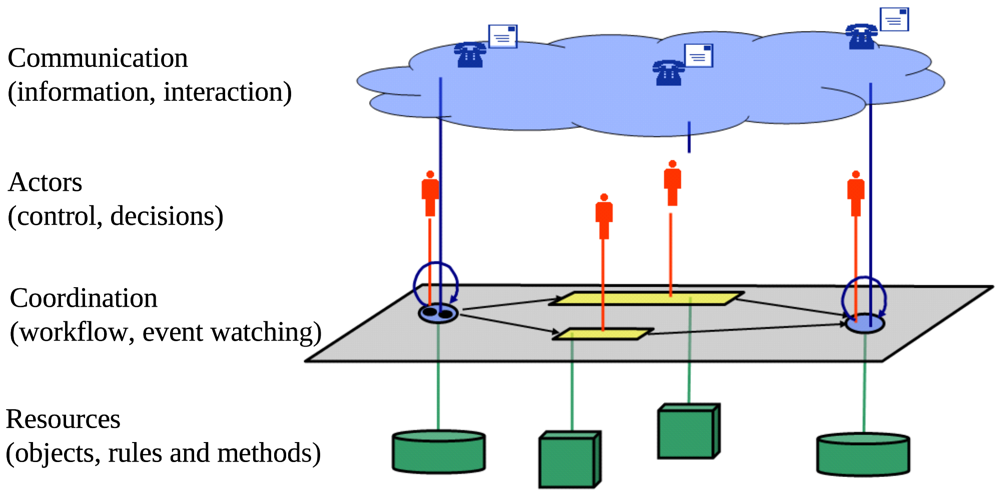


*FIG. 1: Integrative Process Model for Civil Engineering.*

It includes four layers:

- The communication layer, modelling 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 (Rueppel et al., 2002a).
- The actor layer, modelling planners and organisations involved in the process, controlling models and decision making.
- The coordination layer, representing the flow of work within planning. Petri-Nets (Petri, 1962) supply a suitable theory to model this processes (Katzenbach et al., 2002).

- 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 paper focuses on the realisation of the coordination layer and the resource layer. Software agents are used to integrate planning information and knowledge-based services in the process model. The next section gives an introduction of software agents. Afterwards, Petri-Nets are presented as a process modelling method, then the combination of Petri-Nets with software agents in terms of cooperation support is presented.

## 2.3 Software Agents

In the presented approach software agents are used to support model-based cooperation in a distributed planning environment. A software agent is an artificial agent which operates in a software environment. Software environments include operating systems, computer applications, databases, networks and virtual domains. Software agents have the characteristics of delegacy, competence and proactive behaviour. Delegacy stands for the discretionary authority to autonomously act on behalf of the client. These actions include making decisions and performing tasks. Competence is the capability to manipulate the problem domain environment effectively in order to accomplish the prerequisite tasks. This includes specialised communication proficiency. Proactive behaviour is the ability to adapt behaviour to reach a goal.

Software agents reduce human workload by making decisions, acting independently in their environment and interacting only with their end-users when it is time to deliver results. Competence within a software environment requires knowledge of the specific communication protocols of the domain. Protocols, such as SQL for databases, HTTP for the WWW and API calls for operating systems must be implemented into the software agents. Proactive behaviour for software agents is generally limited to providing control options and the generation of status reports that require human review.

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

Mobile agents, also referred to as travelling agents, transfer their code and state between platforms. This often improves performance by moving the agents to where the data resides instead of moving the data to where the agents reside. The typical operation involves a client-server model. In this case, the agent as the client, requests the server to transmit volumes of data back to the agent to be analysed. In many cases, the data must be returned to the server in a processed form by the agent. Load-balancing can be achieved through mobile agents by distributing them over a finite number of computational resources. Some mobile agents are self-distributing, seeking and moving to agent platforms that can offer higher computational resources at lower costs. Further, a complex problem solution can be achieved by dividing the problems into less complex sub-problems with collaborative agents.

Collaborative agents interact with each other to share information or specialised services to effect synergy. They generally share a common interface language which enables them to request specialised services from other agents (Brenner et al., 1998). This means that agents have to communicate. The communication and cooperation with other agents, machines or people is realised by specific languages, such as KQML (Finin et al., 1994) (Labrou & Finin, 1997), ACL (FIPA, 2002) or Tcl (Oustershout, 1994). For the representation of the dynamics the Unified Modelling Language (UML) (OMG, 2001) is expanded to the Agent-UML (AUML) (Odell, 2000).

## 2.4 Agent-Based Model Integration on Example of Fire Protection Planning

### 2.4.1 Introduction

In the presented approach agents are used to integrate the planning models: they are used as “intelligent” and mobile model-interfaces. To develop such an agent support the specialities of the planning activities and the resulting demands on the corresponding technical models have to be analysed. In the following this is presented on example of fire protection planning.

Huge fire disasters have caused fire protection to become one of the central aspects in building design. In Germany fire protection is divided into two domains: preventive and defensive fire protection. Preventive fire protection contains all structural, technical and organisational aspects. Fire fighting and rescue services are aspects of defensive fire protection. Within the preliminary planning, the elementary requirements for effective rescue and optimal fire-fighting are created by preventive fire protection (Schneider & Lebeda, 2000). For that purpose, the building geometry and the adjustment of escape routes are important. Furthermore, the requirements on the building components are determined in terms of fire resistance. All measures to guarantee fire protection in a building contribute to the planning objectives which improve the safety level of a building (Klingsohr & Messerer, 2002). The realisation of the defined protection objectives and the specification of fire protection elements in the detailed planning are integral components of a holistic fire protection concept (Löbbert et al., 2000). According to the type and size of a project, planners from different fields (e.g. structural analysis, dimensioning, heating and ventilation) are involved. Methods must be provided to validate the planning results in terms of completeness and effectiveness regarding the fire protection objectives. A large number of codes and regulations concerning fire protection of buildings have to be evaluated. The presented approach focuses on the preventive fire protection planning. This part is the basis for defensive fire protection and has to be considered throughout the whole planning and construction phase.

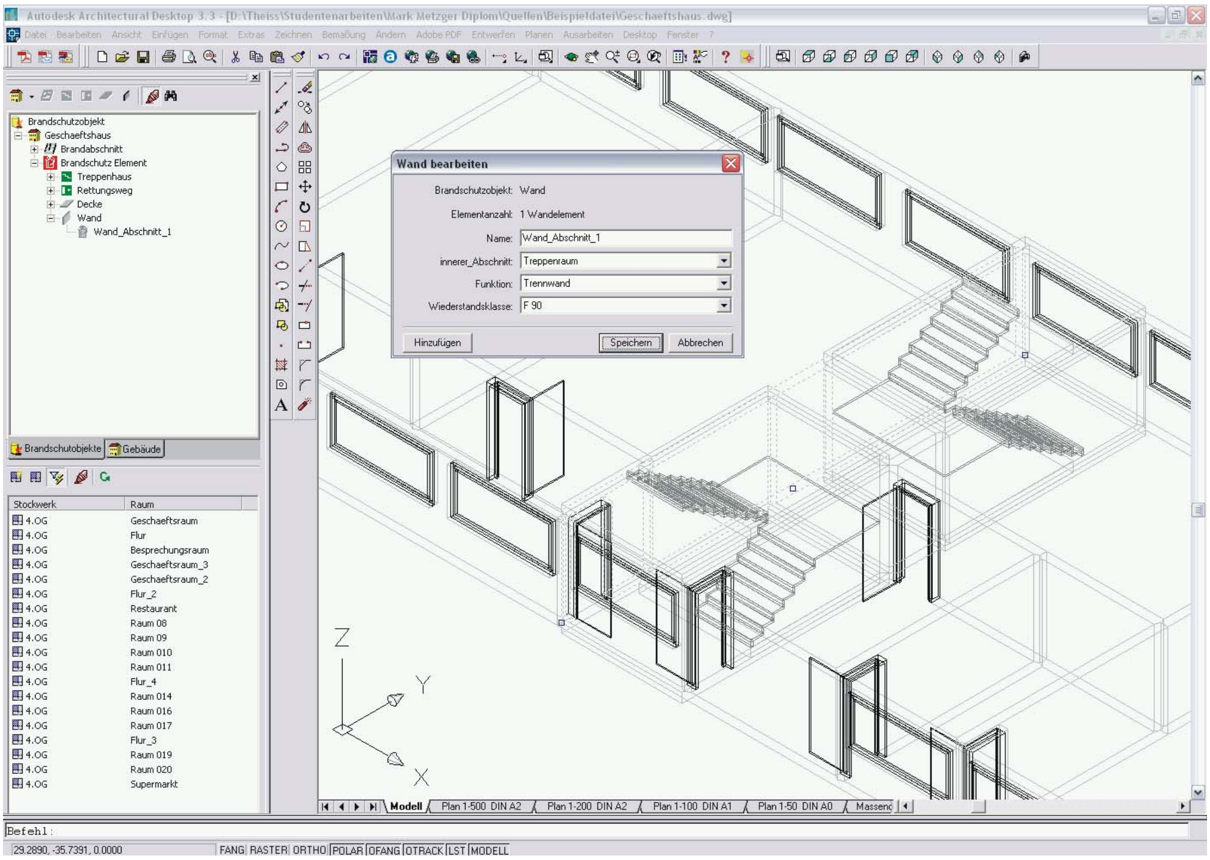


*FIG. 2: Definition of fire protection elements in the CAD model.*

A fire protection concept that describes all planned methods to prevent fire has to be created for every building. In order to process a fire protection concept in a distributed network of planners a transparent model is necessary. Thus, in (Rueppel, et al., 2002a) a new fire protection model has been developed and implemented in the CAD-System Autodesk Architectural Desktop (Fig 2).

During the creation of the fire protection model the new information is directly associated with the building model. A floor in the building model, for example, can be identified explicitly as an escape route. The definition of an escape route implicates special requirements on the linked building components. These requirements must be permanently checked during the planning process. The floor and the walls are already linked together in the building model, so that this connection can be used to check these requirements (Rueppel et al., 2002b). The technical services for facilities have to be designed according to the fire protection model as well, so that planned rescue routes are free of smoke caused by flammable components (Schneider & Lebeda, 2000).

After instantiating the elements of the fire protection model the planning information has to be made available to all planners. Especially planning processes in civil engineering are characterised by the use of many applications with proprietary data formats. For an effective collaboration the data exchange has to be defined in a neutral data format. Consequently, the fire protection model is parsed to XML: firstly, based on the schema specifications of the IAI (IAI, 2002) and secondly, based on a newly developed schema.

### 2.4.2 Approach for Model Integration

The presented approach for cooperation support is based on the assumption that every planner works on a partial model, such as the fire protection model shown in the last section or the structural model. In distributed planning environments the integration of partial models stored in databases is an essential task. In the presented paper a multi-agent-based approach is developed to support model integration (Rueppel et al., 2002a). Each partial 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 (Fig 3). This architecture supports a flexible integration of the partial models.

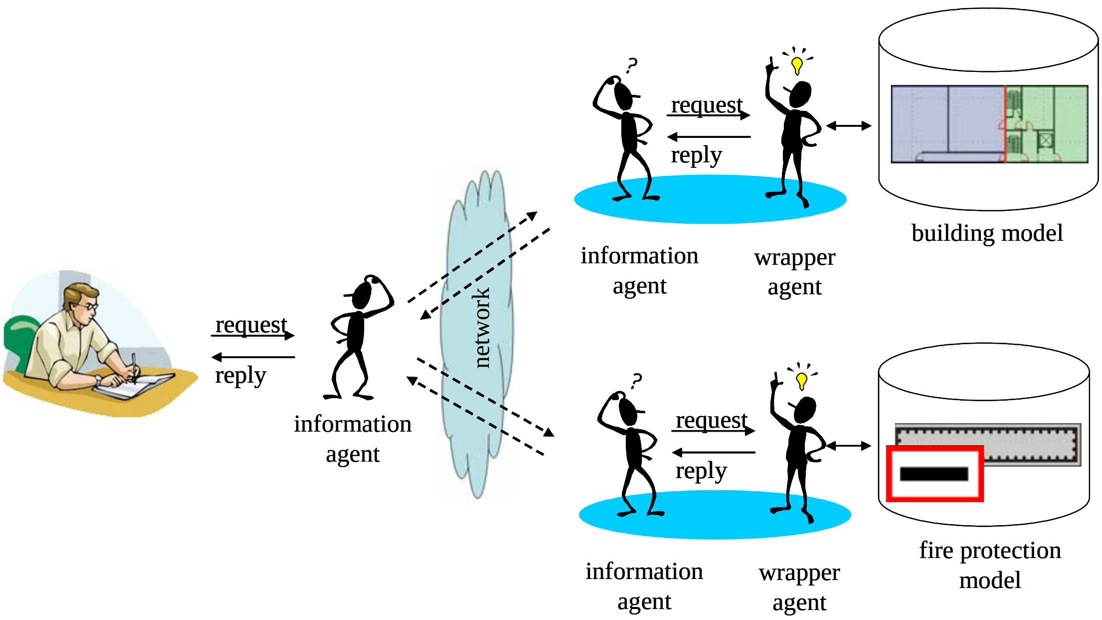


*FIG. 3: Agent-based model integration.*

The database wrapper agents are used to integrate the relevant design information into the multi-agent system. They provide the relevant product model data for other agents in the multi-agent system independent of their physical location. Database wrapper agents therefore 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 to enable mapping of 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 supported (Hartmann et al., 2004).

The described system has been developed on the basis of the Jade-Framework (Bellifemine, 2006), consisting of different agent types which can be instantiated by each planner. Jade (Java Agent Development Framework) is a robust and efficient framework for the development of distributed agent systems. The Jade platform can be distributed to several machines, running different operating systems. It implements the FIPA (Foundation for Intelligent Physical Agents) standard and offers a wide range of services such as an efficient messaging architecture (FIPA, 2005). This architecture comprises interaction protocols, content languages, encoding schemes and ontology support for structuring the messages syntactically and semantically. Jade allows synchronous and asynchronous communication by the use of so called behaviours. Furthermore, it supports the implementation of mobile agents, i.e., agents that migrate across multiple network hosts.

Thus, the agent-platform Jade is an excellent basis for the development of a cooperation platform, supporting interaction, communication and information exchange between distributed planning participants and the integration of heterogeneous resources. It offers a Java API and thus can be integrated with other Java based software components.

#### 2.4.3 Wrapper Agent

The FIPA has incorporated the concept of wrapper agents in its specifications for the design of multi-agent systems. The FIPA software integration specification defines so-called wrapper agents that "agentify" external resources. These wrapper agents provide a public interface to specific external resources although these resources usually cannot be accessed directly from other software agents in the multi-agent system. 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 for accessing information and services of the integrated resources. That means they hide the internal database structure which is required to access the encapsulated data resources. The architecture of the database wrapper agent is introduced in (Theiß & Lange, 2003). The main focus in the development of the wrapper agent is on the communication component. Two important issues have to be considered when modelling the communication between agents: firstly, the structure of the communication process and secondly, the syntax and semantics of the message contents (Rueppel et al., 2005a).

The FIPA has defined so-called agent interaction protocols and communication acts (FIPA, 2002) to support agent communication. An interaction protocol describes a complete communication dialogue between two interacting agents. Fig 4 shows the information-request-interaction-protocol, described in AUML (AUML) (Odell et al., 2000), that is implemented in the agent system.

The shown interaction-protocol models the interaction between the information and the wrapper agent. After receiving the communication act (CA) *"request"* for login, the wrapper agent informs the information agent whether it *"refuses"* or *"agrees"*. After a successful login, a request for model information is passed to the wrapper agent. If the query is understood the wrapper agent processes the query and responds using the CA *"inform"*. The wrapper agents and the querying agents need to implement the CA introduced above in terms of JADE-behaviours on a technical level.

In addition to the communication acts and realised 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 paper, an ontology defining the structure of a building is being implemented (Hartmann et al., 2004). The ontology is provided to the agents by an ontology service, acting as a translator in the agent-based planning environment.

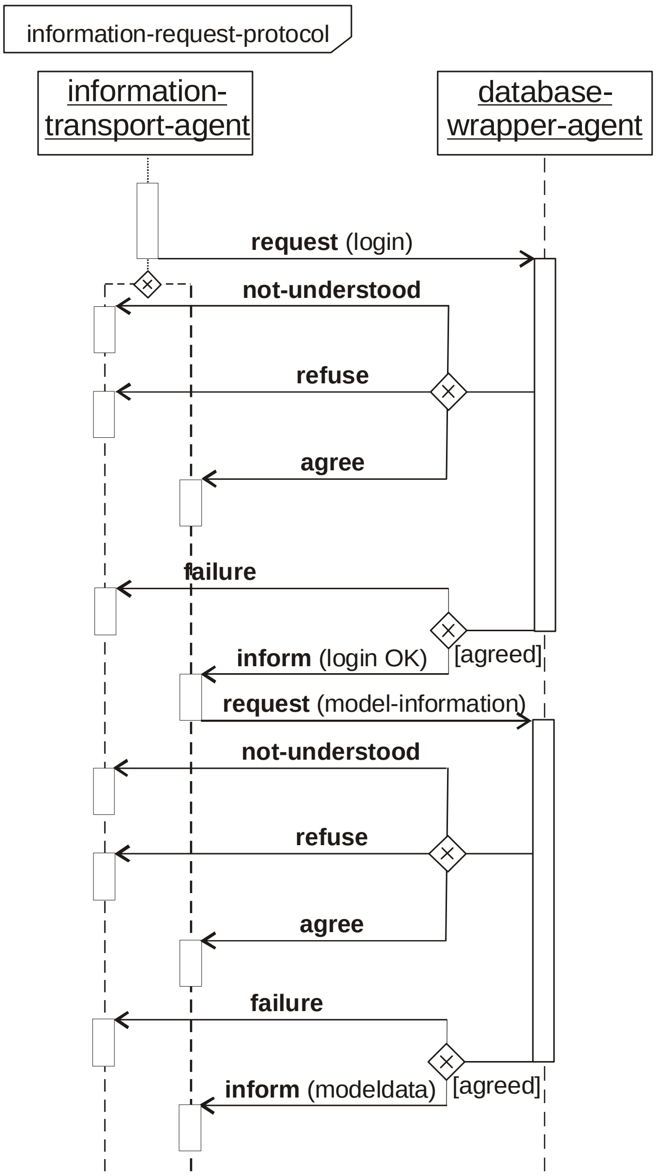


*FIG. 4: Information-request interaction protocol.*

### 2.4.4 Information Transport Agent

As shown above, the different partial product models have to be retrieved by an agent for processing purposes. The information transport agent presented in (Meissner et al., 2004) and outlined in this section offers the service of information transport in the network-based planning environment. The information transport is divided into two steps. The first step is the transport of the query to the database wrapper agent; the second step is the transport of the response from the database to the initiator of the query. Thus, the information transport agent features two main characteristics: mobility and communication.

The multi-agent system consists of several agent-platforms. One platform can comprise several computers and thus a team of planners working in a planning agency. An agent can migrate within this platform. For the inter-platform migration of agents a new service has to be developed (Meissner et al., 2004). Every platform with inter-platform agent migration support has to instantiate this service. The service provides the sending and receiving of agents. The process of agent migration contains several steps: after starting the migration process, the agent class and all inner classes are packed. To restart the agent, all starting parameters have to be saved. Before sending, all data collected on the platform has to be packed as well. The result is then transformed into the content of an Agent-Communication-Language (ACL) message. The content is 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. A multi-agent system with inter-platform-mobility could thus be realised.

The information transport agent uses the same ontology and interaction protocols as the database wrapper agent previously described for communication purposes. The communication is shown in detail in (Hartmann et al., 2004). By the use of the described communication interface, and the shown migration service, an information transport agent is enabled to query and to receive model data. The transport agent analyses the building planner's address and migrates to his platform. On this platform, the agent starts to communicate with the local database wrapper agent and hands out the query for the model data (Fig 3). This query is mapped to the local database schema by the database wrapper agent. The database response is mapped to a public building model ontology and passed on to the waiting information transport agent. This agent migrates back to the initiator with the result. An application example for this agent-based model integration is given later in this paper.

## 2.5 Representing Technical Guidelines

### 2.5.1 Characteristics of Technical Guidelines on Example of Fire Protection Planning

Fire protection, as well as building design supervision, is the sovereign right of the Federal States in Germany. Their codes differ in detail from state to state. The result is a great number of design codes and special building regulations (for multi-storey buildings, hospitals, etc.) (Rueppel et al., 2002a). To comply with these complex standards completely and correctly during the design process, is a challenge in collaborative fire protection planning (Meissner et al., 2004).

A lot of different public and private institutions publish additional fire protection guidelines and regulations, which leads to a great variety of different rules and regulations. Moreover, their contents overlap, complete or replace each other. Furthermore, most fire protection rules do not only have an impact on a special craft but affect many crafts in the building planning process.

Nevertheless, all parties in the building planning process have to work with the complex guidelines and regulations, spending time and money. Fire protection handbooks, best practice manuals, electronic full-text-search and indexing may support the planner in considering the important fire protection aspects, but for a particular planning task they are not sufficient to find the relevant regulations and rules.

In order to support the planner in applying the appropriate rules and guidelines, two knowledge-based services have to be provided: firstly, a human centered representation for a problem oriented search for fire protection guidelines has to be provided in the planning phase. This service structures the complex contents of fire protection guidelines in a semantic term network. Secondly, a computer readable representation of the guidelines for processing in an expert system to provide an automated check of the building model to validate the building design is necessary. The approaches for both services will be presented in the next sections.

### 2.5.2 Term Network to Structure the Contents of Fire Protection Guidelines

In the previous section the particularities of fire protection regulations and guidelines were pointed out. Due to their characteristics it is difficult and time-consuming for the planner to consult all different regulations in order to comply with them. To make the search process in regulation documents more effective, a network of fire protection terms, based on meta-information of the contents of fire protection regulations, has been presented in (Rueppel et al., 2005b). The requirements on the structure of a term network will be analysed in the next subsection. A metadata framework is necessary to implement the term network.

The term network can contain nodes which represent terms of elements of structural fire protection and terms of elements of the building model (e.g., firewall, storey, fire area or emergency stairway) (Fig 5). These terms can be assigned to paragraphs of the regulation documents which contain fire protection rules about the elements the terms stand for. The term "firewall", for example, can be assigned to a paragraph in which the fire resistance class of a firewall is defined. In most cases the description of a paragraph by a single term is too general to achieve precise search results. Consequently, it is possible to assign two or even more terms to a particular paragraph.

A further requirement on the term network results from the following: according to a special paragraph in the fire protection regulations, a firewall of a central heating room located in the basement has to conform to a high fire resistance class. The terms “firewall” “central heating room” and “basement” could be assigned to this paragraph. Additionally, simple nexuses between terms, semantic connections – in form of verbs – should be possible to augment the expressiveness of the term network. One example of a semantic connection could be the expression “has a” between the term “emergency stairway” and the term “window”.

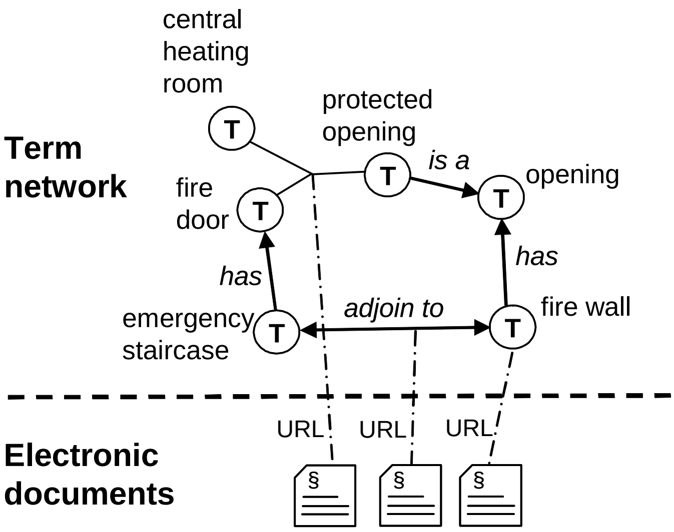


*FIG. 5: Structure of the term network.*

As planners of different disciplines have to consult fire protection regulations, they often use their own terminology and have their own perception in mind. In order to take this aspect into account synonyms are represented in the term network.

In order to implement a term network for fire protection regulations, different metadata frameworks are offered. The Topic Maps framework (Pepper & Moore, 2001) entirely meets the aforementioned requirements and was chosen to represent the semantic term network in this project.

The most important element of the subject-centric standard is the topic itself. It has a name and can be a representation of any physical or non-physical subject. In the case of the developed term network it is used to model the fire protection terms. Topics can also have one or more topic types which allow the modelling of hierarchical relations like a “firewall” being a “building component”. The topics can be interlinked by associations. An association does not have an implicit direction but can be determined by either the human reader or by the role that the two topics play in the association. Apart from associations between two topics, n-array associations can also be modelled with Topic Maps.

Using the described elements, a network of metadata can be built expressing coherences between the different contents of fire protection regulations. The connection of the network to the electronic resources is realised by occurrences. Occurrences are available to assign associations and topics to the underlying information resources (electronic documents). Another important element in Topic Maps is the scope. Topics, associations and occurrences can have an unlimited number of scopes which enable multiple contexts. Existing synonyms in different regulations and technical guidelines dealing with fire protection issues can be represented by scopes for the names of the topics in the term network.

It is also possible to assign the characteristics (names, associations and occurrences) of a topic to associations and occurrences using the technique of reification. This means, creating a topic which does not represent a subject of the real world but rather an element in the Topic Map itself. As a result, the characteristics of this topic also belong to the Topic Map element (association or occurrence) it reifies. In the presented approach, this technique is mainly used to name associations and to assign not only topics to paragraphs of electronic documents but is also used to assign associations to the electronic resources.

During the search the planner is supported by a graphical user interface which represents the term network. The visualisation allows the planner to navigate in the network and finally to display the electronic documents which are assigned to the network elements. Thus, the planner is able to specify his planning task by navigating through the network and then receives a problem specific list of guidelines he has to consider.

This service is offered as part of a web-based planning environment described below and is used in the planning phase. A second service is presented in the next section. It supports the planner by an automated check of the building design for compliance with an essential set of fire protection regulations in the pre-construction phase.

### 2.5.3 The Fire Protection Agent

The second service offers "semi-automated" checks of the structural elements for compliance with the fire protection regulations. This service is offered by an "intelligent" fire protection agent.

The definition of a pure data model is not sufficient to check the functionality of planning information for fire protection (Meissner et al., 2003). The information defined in a fire protection concept is associated to the rules of codes and guidelines which must also be modelled. The processing of these rules in an expert system to check the compliance of the building model with the fire protection guidelines, can be realised by the efficient combination of model data and rules. A three-tier knowledge model for preventive fire protection was therefore developed: Firstly, it consists of the data models for the building; secondly, a rule model was developed to represent the guidelines and regulations of fire protection and thirdly, it consists of the fire protection model for the building.

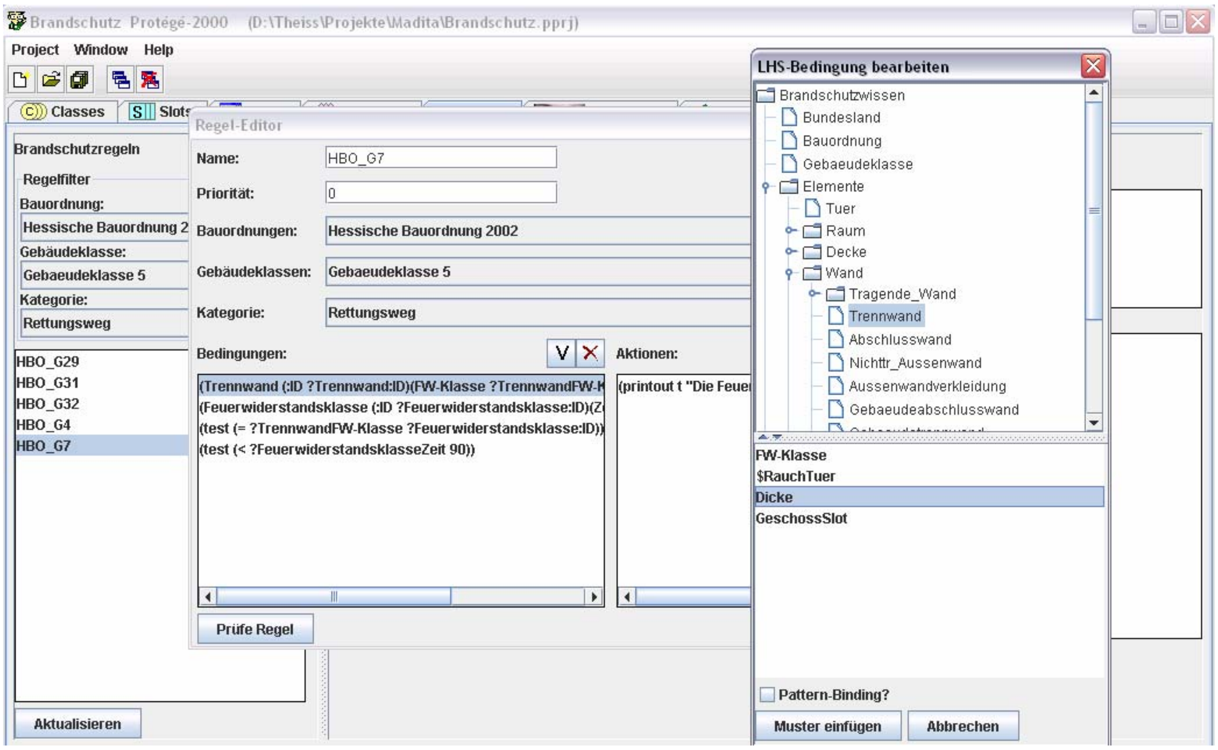


*FIG. 6: Editor for defining rules of fire protection element validation.*

For the application of fire protection rules to the data models, the rules must be defined and structured in a processible form. The rules are thus structured in a first step by the type of the regulation and in a second step by the related building element. This approach enables an object-oriented processing of necessary rules. The rule structure optimises the validation process as well as the communication process for acquiring necessary rules.

By means of a problem-specific user interface (Fig 6) the CLIPS-based rules (Riley, 2002) can be defined in a graphical editor, implemented on the basis of Protégé (Protégé, 2006). Thus, they can be processed and integrated in the structure described above (Theiss, 2002). Complex rules can be set up conveniently by the combination of several rules.

A “fire protection agent” has been presented in (Rueppel et al., 2002a) for this purpose. This agent has to be able to process the facts of the building’s fire protection model according to the requirements from the design codes. The rules in fire protection are declaratively styled. Rule-based expert systems are well known in processing declarative rules. The Java Expert System Shell (Jess) (Friedmann-Hill, 2003) is one of the most common rule-based expert systems and the reference implementation for the Java Rules Engine API. The Jess API can be integrated directly in an agent of the Jade system. Rule-based systems work with rules and facts. 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.

The fire protection agent has to process information from all models involved (Meissner et al., 2003) (Fig 7). Firstly, the elements which are to be checked have to be identified in the fire protection model. Secondly, according to the above described rule structure all relevant rules have to be retrieved from the fire protection rule model by the information transport agent and the corresponding database wrapper agent. Thirdly, the corresponding planning model elements have to be retrieved from the building model databases in the same way. After retrieving all the information, the fire protection agent checks whether it has all the facts to process the rules. This is an iterative process. So, step by step, all rules and facts needed for the fire protection requirement check, are transported to the fire protection agent. As a result, the fire protection elements can be checked for compliance with the valid fire protection regulations by the built-in inference engine (expert system). This approach enables the planner to check his design for compliance with the fire protection regulation, supported by “intelligent” agents.

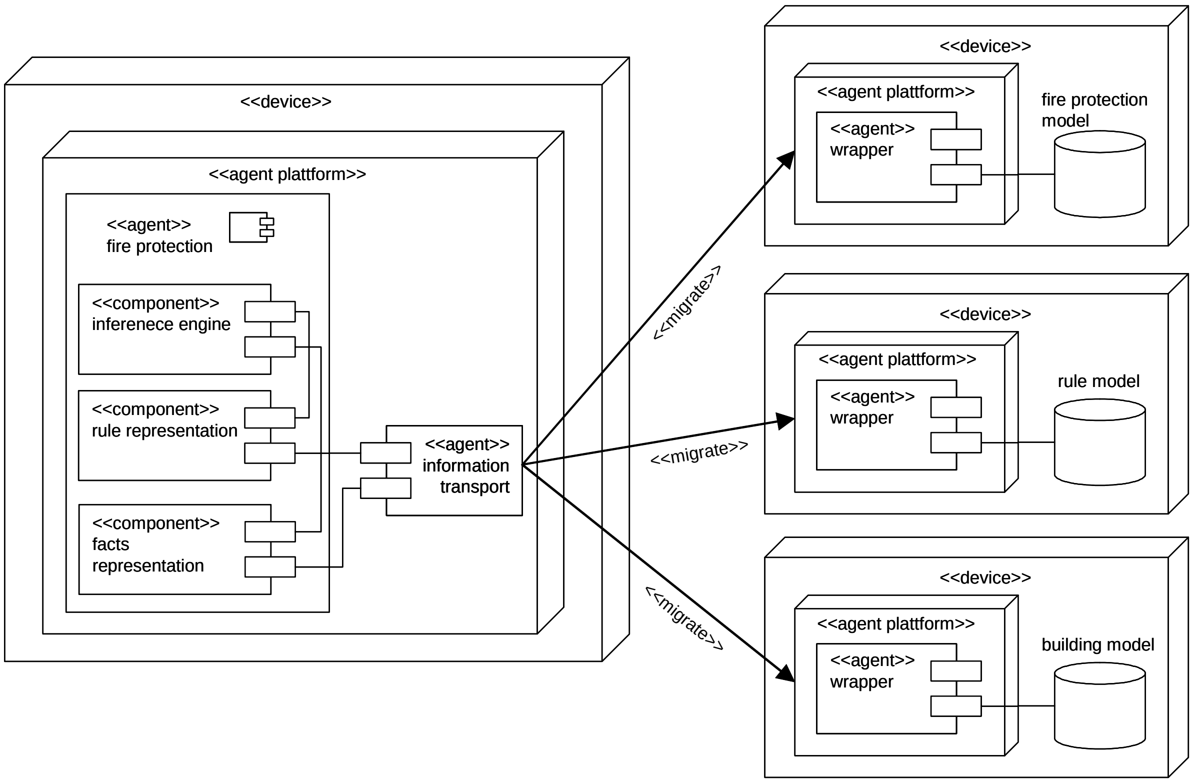


*FIG. 7: Agent-based check for compliance with fire protection regulations.*

### 2.5.4 Fire Protection Planning Environment

As described above, the discipline of fire protection planning includes planners from many technical fields. The fire protection engineer defines objectives and measures in a fire protection concept. All other planners have to consider the requirements of fire protection during their detailed planning. The aim of this part of the presented approach is to provide all planners with the developed knowledge-based services. As described above the fire protection partial model is highly interrelated to other partial models. One of the planning process’ main goals is to build up these partial models while preserving consistency among each other. The integration of the distributed partial models is demonstrated in section 2.4. Based on the integrated partial models, knowledge-based services are implemented as described before. These services are the core of the planning environment as presented in (Rueppel et al., 2005a) and shown in Fig 8.

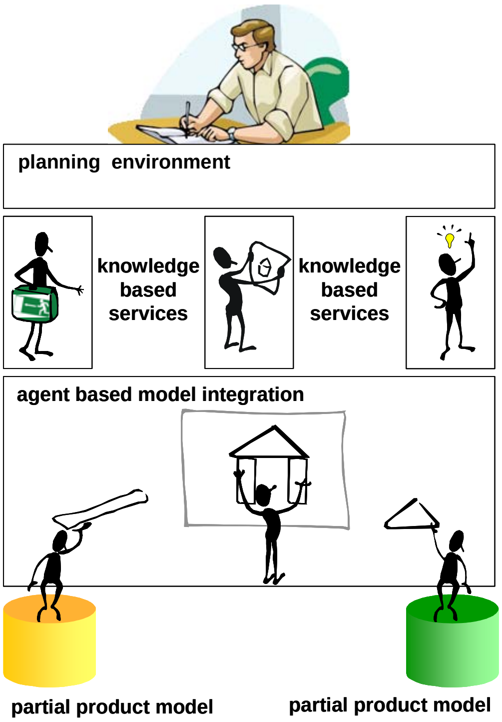


*FIG. 8: Agent-enabled model integration and knowledge-based services as a planning environment.*

A web-based user interface for the knowledge-based planning environment has been implemented (Fig 9). This interface allows the planner to access the distributed partial models and different knowledge-based services and provides different views of the fire protection elements of a building.

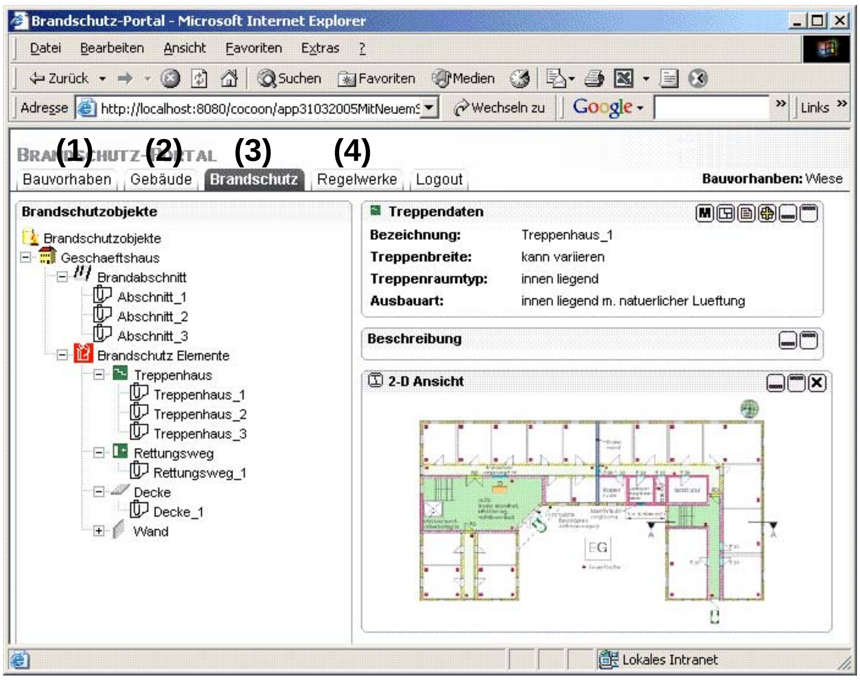


*FIG. 9: Web interface of the knowledge-based planning environment with a tree view and a 2D view of the fire protection elements (in German language).*

This environment offers:

1. information about the building project,
2. general building data,
3. different views on the fire protection model and related information (automatically integrated by agents), and
4. the knowledge-based services concerning fire protection guidelines.

## 2.6 Process Modelling With Petri-Nets

The implementation of the resource layer of the integrative process model was described in the last sections. An adequate modelling method is required to realise the coordination layer of the integrative process model. Petri-Nets supply a theory for the modelling and control of concurrent and asynchronous processes in a distributed environment. In (Katzenbach et al., 2002) the characteristics of the planning processes in Civil Engineering and the assessment of different modelling methods are presented. Based on these characteristics, the requirements on the process modelling method were deduced. Considering these requirements, different modelling methods were analysed, such as network plans, the Unified Modelling Language (UML) and Petri-Nets. The latter has proved to be a powerful process modelling method. Petri-Nets provide a mathematical formalism for the definition and representation of a discrete system, e.g., (Kusiak & Yang, 1993).

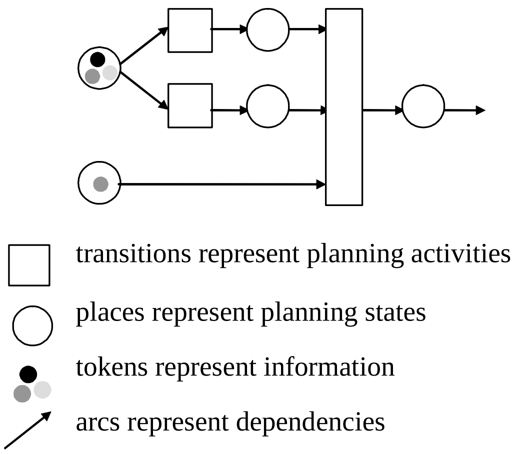


*FIG. 10: Process Modelling with Petri-Nets.*

By means of tokens the dynamic behaviour of a system is enabled. Petri-Nets with individual tokens in particular (Coloured Petri-Nets, Predicate-Transition nets or High Level Petri-Nets) allow the use of semantic information for decision modelling. The Petri-Net theory, originally developed by Carl Adam Petri (Petri, 1962), is well known in computer science. Throughout the years further contributions have extended the original method, e.g., (Jensen, 1996). In the late 90s, van der Aalst (Aalst, 1998) and Oberweis (Oberweis, 1996) introduced the application of Petri-Nets to process modelling.

Petri-Nets basically consist of places (p) and transitions (t). Places and transitions are connected by directed arcs. Places can hold tokens. In Coloured Petri-Nets additional information can be attached to each token. The basic idea in modelling planning processes with Petri-Nets, is to model planning states by places, planning activities by transitions, relations between planning states by arcs and information by (coloured) tokens. Fig 10 shows the elements of a Petri-Net and a sample net. The significant reasons for modelling civil engineering processes with Petri-Nets are:

- the graphical representation,
- the bipartite structure with places and transitions for modelling both planning states and planning activities,
- the token concept for modelling logical conditions and the flow of planning information, and
- the mathematical formalism for structural, behavioural and simulation analysis of engineering process models.

## 2.7 Cooperation Support

To realise cooperation support the above presented methods for the resource and coordination layer of the integrative process model are combined in the following way: firstly, wrapper and information agents are used to integrate the distributed planning resources; secondly, the fire protection agent provides knowledge-based methods to process this information for model-checking semi-automatically and thirdly, Petri-Nets are used to coordinate the workflow of the planners and corresponding agents.

The combination of agent oriented methods and the Petri-Net method is an ongoing research topic in computer science. Various approaches exist to combine these methods, each emphasising different aspects. In (Xu & Deng, 2000) for example, the modelling of mobile agent systems with High Level Petri-Nets is proposed. Hereby, the agent system is represented by a Predicate-Transition Net and the individual agents are modelled as tokens. Agents migrate from one component to another by transition firing at runtime. In (Nowostawski et al., 2001) or (Cost et al., 1999) a formalism based on Petri-Nets that is to be used for modelling the complex and concurrent conversation between agents and multi-agent systems is shown. Finally, in (Moldt & Rölke, 2003) the focus lies on modelling the internal behaviour of an agent by Coloured Petri-Nets. For example, the two different behaviour patterns of an agent, namely “reactive” or “proactive”, are modelled as concurrent transitions of a Petri-Net.

The presented approach is based on two net types (Rueppel et al., 2003b). The first net type contains, the planning activities, planning states, planning information and the sequential, iterative or concurrent dependencies. This net is referred to as the “planning net”. Within this net the transitions represent planning activities or events. 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 or instances during runtime. As the focus lies on the agent modelling, this net is referred to as the “agent net”.

The combination of these two net types enables a holistic modelling of the agent-based support of planning processes with a unique semantic. This is shown in a planning scenario in the next section.

# 3. EVALUATION SCENARIO IN FIRE PROTECTION PLANNING

To evaluate the presented methods, a fire protection requirement check is modelled. The example describes the design of a fire break wall after adding two fire compartments to the building model. The fire compartment separates a section with offices from a section with housing rooms (Fig 11). This design task is based on a close communication and collaboration between the architect, the structural engineer and the fire protection engineer.


*FIG. 11: Design of Fire Protection Elements.*

*TABLE 1: Typical planning participants, activities, states and information in a fire protection planning scenario.*

| | |
|---|---|
| Planning participants | |
| AC | Architect |
| SE | Structural Engineer |
| FPE | Fire Protection Engineer |
| | |
| Planning activities | (transitions ti) |
| t1: | AC requests the fire protection concept |
| t2: | FPE defines fire compartment |
| t3: | FPE checks fire protection concept |
| t4: | FPE sends information to AC |
| t5: | FPE sends information to AC |
| t6: | SE designs fire break wall |
| t7: | SE dimensions the fire break wall |
| t8: | SE sends information to FPE |
| | |
| Planning states | (places pi) |
| p1: | necessity of the fire protection concept |
| p2: | building model with one fire compartment |
| p3: | building model with two fire compartments |
| p4: | fire protection concept is OK |
| p5: | building model with fire protection information |
| p6: | fire protection concept is NOT ok |
| p7: | building model with wrong fire break wall |
| p8: | building model with request for re-design the fire break wall |
| p9: | fire break wall is dimensioned |
| p10: | server with fire protection rules |
| p11: | server with structural design norms |
| | |
| Planning information | (tokens) |
| | building model |
| | fire protection rules |
| | structural design norms |

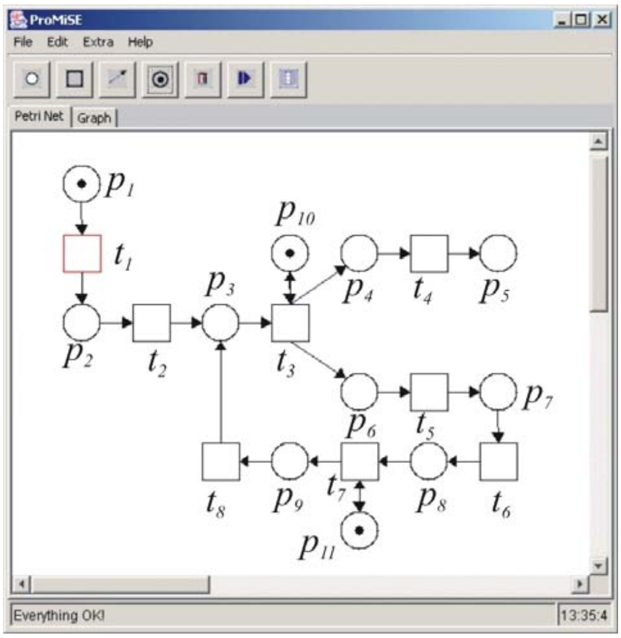


*FIG. 12: Planning scenario in fire protection planning modelled as C/E net in PROMISE (planning net).*

Table 1 provides the planning participants, the planning activities, the planning states and the information. In figure 12 the developed Petri-Net server PROMISE is shown (Rüppel et al., 2003a). This sever enables the interactive modelling, analysing and processing of Petri-Nets and is part of the knowledge-based planning environment. Figure 12 provides the concurrent and iterative design processes to include the fire protection concept into the building model.

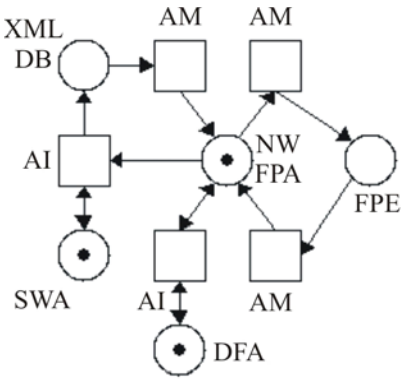


*FIG. 13: Agent migration and interaction modelled as a Petri-Net (agent net).*

In this scenario the planning activity "FPE checks fire protection concept", represented by transition t3, is associated to a fire protection agent. This fire protection agent instructs the information agent to migrate through the network and to communicate with different stationary wrapper agents. In doing so, the information agent uses a directory service, requests information from a XML database and finally carries the information back to its source location (i.e. the workstation of the fire protection engineer, where the fire protection agent can check the planning information for compliance with the fire protection regulations). These processes are modelled in the agent net and illustrated in Fig 13. The abbreviations, defined in table 2, are used in the agent net.

*Table 2:. Abbreviations.*

| | |
|---|---|
| FPE | fire protection engineer |
| FPA | fire protection agent |
| DFA | directory facilitator agent |
| SWA | stationary wrapper agent |
| NW | network |
| XML-DB | XML database |
| AM | agent migration |
| AI | agent interaction |

With the combination of the planning and agent net all necessary steps concerning the re-use of the building and the check of fire protection requirements are modelled for semi-automated processing. All previously presented software-modules of the knowledge-based planning environment are operated by the nets.

## 4. CONCLUSION

Development of appropriate software methods in order to organise and support distributed engineering processes in computer networks is an ongoing research area. The presented integrative process model is an appropriate basis for the development of software for the network based cooperation support in civil engineering.

The coordination layer of the integrative process model is realised by the use of Petri-Nets. Petri-Nets supply a theory for the modelling and control of concurrent and asynchronous processes in a distributed environment. On the one hand, the formal modelling of planning states by places and planning activities by transitions is complex and not intuitive. On the other hand, Petri-Nets allow a precise description of the planning processes, guarantee well-defined syntax and semantics and enable the analysis of the process models in order to avoid modelling mistakes.

The integration of models and knowledge based services in the resource layer was realised by the use of software agents. The used agent development framework Jade offers a variety of essential services to support the development of a distributed planning environment for civil engineering. The comprehensive communication capabilities and the support of mobility enable a flexible and efficient implementation of a cooperation platform supporting interaction, communication and information exchange between distributed planning participants.

The presented combination of software agents and Petri-Net-based process modelling has turned out to be an appropriate approach to integrate planning resources with the process model and thus connecting the coordination to the resource layer of the integrative process model. Thus, model based engineering methods and legacy software systems can be integrated with process oriented approaches to support distributed planning processes in civil engineering. The presented results enable a high quality and efficient cooperative project work with distributed technical models in networks.

## 5. REFERENCES


van der Aalst W. M. P. (1998). The Application of Petri-Nets to Workflow Management. *Journal of Circuits, Systems and Computers* 8(1), World Scientific, Singapur. 21-66.

Bellifemine F. (2006). *Jade - Java Agent Development Framework*, TiLab, Turin, Italia. (http://jade.cselt.it)

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

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

Brenner W., Zarnekow R. & Wittig, H. (1998). *Intelligente Software-Agenten*, Springer, Berlin, Germany.

Cost R., Chen Y., Finin T., Labrou, Y. & Peng, Y. (1999). Modeling Agent Conversations with Colored Petri Nets. In: *Proc. Third Conference on Autonomous Agents (Agents-99), Workshop on Agent Conversation Policies*, Seattle, WA, USA.

Finin T., Fritzson R., McKay D. & McEntire R. (1994). KQML as an Agent Communication Language, In: *Proceedings of the third International Conference on Information and Knowledge Management (CIKM94)*, ACM Press, USA 1994.

FIPA (2002). *ACL - Agent Communication Language*, Foundation for Intelligent Physical Agents (FIPA) specifications homepage. (http://www.fipa.org)

FIPA (2005). *Foundation for Intelligent Physical Agents*. (http://www.fipa.org)

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

Hartmann D., Meissner U. F., Rueppel U., Bilek, J. & Theiss, M. (2004). Integration of Product Model Databases into Multi-Agent Systems. In: *Proc. of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)*, Weimar, Germany.

Hollingsworth D. (1995). *Workflow Management Coalition (WfMC) –The Workflow Reference Model*. Document Number WfMC-TC-1003. (www.wfmc.org)

Huhnt W. (2004): Progress Measurement in Planning Processes on the Base of Process Models. In: *Proc. of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)*, Weimar, Germany.

IAI - International Alliance for Interoperability (2002). *ifcXML-Schema Definition - Version 1.02*. (http://www.iai-international.org)

Jensen K. (1996). *Coloured Petri-Nets – Basic Concepts, Analysis Methods and Practical Use*, Springer-Verlag, Berlin, Germany.

Karhu Vesa (2001): A Model Based Approach for Construction Process Modelling. In: *Proceedings of the CIB-W78 International Conference: IT in Construction in Africa 2001*, Mpumalanga, South Africa.

Katzenbach R., Meissner U. F., Rueppel U., Giere, J. & Greb S. (2002). Process-oriented Network-based Collaboration in Geotechnical Engineering. In: *Proceedings of the 9$^{th}$ International Conference on Computing in Civil and Building Engineering Vol. II,* Taiwan, 1063-1068.

Klingsohr K. & Messerer J. (2002). V*orbeugender baulicher Brandschutz*. Stuttgart: Verlag W. Kohlhammer.

König M., Klinger A. & Berkhahn, V. (2004). Process Modelling in Building Engineering. In: *Proc. of the 5th European Conference on Product and Process Modelling in the Building and Construction Industry (ECPPM),* Istanbul 2004.

Kusiak A. & Yang H.-H. (1993). Modeling the design process with Petri-Nets. In: *Concurrent Engineering, Contemporary issues and modern design tools*, Chapman & Hall, London, UK, 447-464.

Labrou T. & Finin T.(1997). *A Proposal for a new KQML Specification,* In: Technical Report TR CS-97-03, University of Maryland, USA.

Löbbert A., Pohl K. D. & Thomas, K.-W. (2000). *Brandschutzplanung für Architekten und Ingenieure*. Köln: Verlag Rudolf Müller.

Meissner U. F., Rueppel U. & Theiss M. (2004). Network-Based Fire Engineering Supported by Agents. In: *Proc. of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X),* Weimar, Germany.

Meissner U. F., Rueppel 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. (2002). Verteilte Brandschutzmodellierung auf der Basis von Software-Agenten. In: *Bauen mit Computern - Kooperation in IT-Netzwerken*, Bonn, Fortschritt-Bericht VDI, 557-569.

Moldt D. & Rölke H. (2003). *Verhaltensmodellierung von Petrinetz-Agenten*.

Nowostawski M., Purvis M. & Cranefield S. (2001). Modelling and Visualising Agent Conversations. In: Muller J. P., Andre E., Son S. and Frasson C. (eds.) *Proc. of the 5th International Conf. on Autonomous Agents*, New York, ACM Press. 234-235.

Oberweis A. (1996). *Modellierung und Ausführung von Workflows mit Petri-Netzen,* Teubner-Verlag, Stuttgart, Germany.

Odell J., Parunak H. V. D. & Bauer B. (2000). Extending UML for Agents. In: *Proc. of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence*.

OMG - Object Management Group (2001). *Specification of the Unified Modelling Language (UML)*, Vers. 1.4.

Oustershout J. K. (1994). *Tcl and the Tk Toolkit*, Addison Wesley, New York, USA.

Pepper S. & Moore G. (2001). XML Topic Maps (XTM) 1.0 TopicMaps.org Specification. (http://www.topicmaps.org/xtm/index.html).

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

Protégé (2006). *Ontology Editor and Knowledge-base Framework*. Stanford Medical Informatics (http://protege.stanford.edu).

Riley G. (2002). *C Language Integrated Production System (CLIPS) - A Tool for building Expert Systems*. (http://www.ghg.net/clips/CLIPS.html).

Rueppel U., Greb S. & Theiss, M. (2003b). 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*, Vol. 2, 651-656, Madeira Island, Portugal.

Rueppel U., Lange M. & Wagenknecht A. (2005b). Structuring Technical Guidelines in Fire Protection Engineering with Topic Maps. In: Scherer R. J., Katranuschkov P. & Schapke S.-E. (Eds.) P*roc. of the 22nd CIB-W78 Conference on Information Technology in Construction*, CIB Publication No. 304, ISBN 3-86005-478-3, Dresden, Germany, p. 335-341.

Rueppel U., Meissner U. F. & Theiss M. (2002a). Fire Protection Concepts With Mobile Agents. In: *Proc. of the 9th International Workshop of the European Group for Intelligent Computing in Engineering (EG-ICE)*, Darmstadt, Germany.

Rueppel U., Meissner U. F. & Theiss M. (2002b). An Agent-based Platform for Collaborative Building Engineering, In: *Proc. of the 9th International Conference on Computing in Civil & Building Engineering*, Taipei, Taiwan.

Rueppel U., Theiss M. & Lange M. (2005a). Agent-enabled Model Integration in a Knowledge-based Planning Environment. In: Scherer R. J., Katranuschkov P. & Schapke S.-E. (Eds.) P*roc. of the 22nd CIB-W78 Conference on Information Technology in Construction*, CIB Publication No. 304, ISBN 3-86005-478-3, Dresden, Germany, p. 471– 476.

Rump F. J. (1999). *Geschäftsprozeßmanagement auf der Basis ereignisgesteuerter Prozessketten*. Teubner-Verlag.

Rüppel U., Meißner U. F. & Greb, S. (2003a). Vernetzte Bauprozesssteuerung auf der Basis bauteilspezifischer Prozessmuster für geotechnische Konstruktionselemente. In: Gürlebeck K., Hempel L. & Könke C. (Hrsg.): *digital Proceedings, 16. Internationales Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen* (ikm 2003; Tagungs-CD-ROM), ISSN 1611-4086, Weimar.

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

Theiß M. & Lange M. (2003). Integration von XML-basierten Fachinformationen in Multiagentensysteme. In: Kaapke K. & Wulf A. (Hrsg.): *Forum Bauinformatik 2003 - Junge Wissenschaftler forschen*. Hannover: Shaker Verlag.

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

Xu D. & Deng Y. (2000). Modeling mobile agent systems with High Level Petri Nets. In: *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'2000)*, 8-11 October 2000, Nashville, NT, USA, p. 3177-3182.