

Collaborative Fire Protection Engineering Based on Software Agents

by

Uwe Rueppel¹

ABSTRACT: This paper presents a new approach with software agents to support the task of preventive fire protection engineering. Agents are stationary or mobile software components to support the user in his planning tasks according to their specific design. Agents can act autonomously within their environment, which means that they can solve problems without the need of control through a human being or another system. In order to react to modifications of the environment agents observe their environment continuously and react adequately to modifications. To reach a result it is necessary for autonomous agents to react particularly in a goal-oriented way. The purpose of an agent system in the area of fire protection engineering is to organize a distributed problem solution by many agents representing adaptive experts. Using available object-oriented, net-enabled models of standard applications and based on the technology of the software agents, a platform for collaborative fire protection engineering is designed. Local service-agents provide necessary information in a collaboration platform. By communication between these local service-agents and mobile verifying-agents the appropriate information can be processed to verify the fire protection model of a building.

KEYWORDS: Collaboration platform; fire protection engineering; information technology; software agents;

1. INTRODUCTION

Fire protection engineering is a part of the building design process in civil engineering and consists of the cooperation of experts in multiple disciplines. Consistent cooperation of engineers in different fields is the requirement for high product quality, for optimal and short development with a minimum of investment costs. The development process in building engineering is strongly assigned by creating unique buildings. Software methods and tools in building engineering do not sufficiently

support the communication, collaboration and control processes, especially in the area of fire protection engineering. Most software solutions are based on proprietary solutions with their specific technical model. Collaboration is often implemented as a centralized client-server architecture. Model information in these client-server architectures are static with the consequence that every completion of a new model requires the definition of new interfaces and implementations in every module [1].

¹ Professor for Computer Science in Civil Engineering
Darmstadt University of Technology
Petersenstrasse 13, 64287 Darmstadt, Germany
Tel.: +49 (0) 6151 16 34 44
Fax: +49 (0) 6151 16 55 52
rueppel@iib.tu-darmstadt.de

2. Fire Protection Engineering

In addition to the described problems in collaboration, a great number of codes and regulations for fire protection models of buildings have to be evaluated. In Germany every state has different fire protection codes and a great number of special building regulations (for multi-storey buildings, hospitals, etc.) (see Figure 1). To consider these complex standards completely and without errors during the design process is a challenge in collaborative fire protection engineering. Fire protection models therefore are often inconsistent.

German codes define the fire protection needs overall. There are different parts of fire protection engineering:

- Organizational;
- Operational;
- Defensive; and
- preventive.

The focus of the presented approach is the preventive fire protection engineering. This part has to be recognized during the whole planning and construction phase and is the basis for the defensive fire protection. The goals in preventive fire protection engineering are:

- prevention of fire;
- containing the fire using fire compartments;
- developing an infrastructure for extinguishing the fire (fire extinguisher, sprinkler etc.)
- and developing an infrastructure for rescue.

As it is shown in Figure 2, these preventive fire protection elements are represented by or have strong relationships to building elements.

Normally the fire protection model is developed by architects as a part of their building design. Based on this draft, other planners have to complete the detailed planning and advance the fire protection model continuously (see Figure 3). That means, that for example a structural engineer finds restrictions due to fire resistance demands in the choice and design of the structural elements. The technical services for facilities have to be designed according to the fire protection model, so that planned rescue routes

are free of smoke from flammable components [2]. After some great fire disasters in Europe, the compliance and inspection of fire protection rules are taken very seriously. With regard to the devastating consequences to people living and working in buildings, it is necessary to include the fire protection concepts very early in the building design process being conducted by the all specific planners involved.

In order to support the disciplinary processes of the fire protection model evolution in a distributed environment, the integration of all participants in the entire planning process is necessary. Therefore data, methods and knowledge of the partial models are to be coupled for collaboration, while taking into consideration that the planners are spatially distributed. The concept of collaborative work of distributed planners must support sequential, parallel and iterative planning processes within an integrated information platform.

3. Software Agents

An agent is one that is authorized to act for another. Agents possess the characteristics of delegacy, competency and proactive behaviour. Delegacy means the discretionary authority to autonomously act on behalf of the client. Actions include making decisions, committing resources, and performing tasks. Competency is the capability to effectively manipulate the problem domain environment to accomplish the prerequisite tasks. Competency includes specialized communication proficiency. Proactive behaviour is the ability to adapt behaviour to optimize performance in an often non-stationary environment in responsive pursuit of the goals of the client.

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. Delegacy for software agents means that they stay resident or persistent as background processes after being

launched. By making decisions and acting on their environment independently, software agents reduce human workload by generally only interacting with their end-clients when it is time to deliver results. Competency 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.

Mobile Agents, also known as travelling agents, will shuttle their code and state among resources. This often improves performance by moving the agents to where the data reside instead of moving the data to where the agents reside. The alternative typical operation involves a client-server model. In this case, the agent, in the role of the client, requests that the server transmit volumes of data back to the agent to be analyzed. Oftentimes the data must be returned by the agent to the server in a processed form. With mobile agents load-balancing can be achieved by distributing agents over a finite number of computational resources. Some mobile agents are self-distributing, seeking and moving to agent platforms that can offer the higher computational resources at lower costs. Further a complex problem solution can be achieved by dividing the problems into less complex sub-problems addressed by collaborating agents.

Collaborative agents interact with each other to share information or barter for specialized services to effect a synergism. While each agent may uniquely speak the protocol of a particular operating environment, they generally share a common interface language which enables them to request specialized services from other agents as required. This means that agents have to communicate [3]. The communication and cooperation with other agents, machines or people is realized by specific languages as KQML [4] [5], ACL [6] or Tcl [7].

4. Fire Protection Engineering with Software Agents

These qualities of agent systems are utilized for the evolution of an integrated fire protection model. The information of the fire protection model is determined from two different sources. On the one hand, fire protection relevant rules must be accessible to the system. On the other hand, the information of building product models must be available. The fire protection model must be adapted to the different views of the planners within the collaborative building design and relevant information must be evaluated within the agent-based platform [8]. The system architecture of the collaboration platform for preventive fire protection engineering is shown in Figure 4.

The determination and access of the fire protection information are designed as a mobile verifying-agent. The architect designs a fire protection model on the basis of his building design in his working environment. He defines the elements, relevant for preventive fire protection. A fire protection application for AutoCAD ADT was developed for this purpose. On completion of a design, the architect publishes the fire protection model on his agent-platform. For this purpose a fire protection- XML-interface for AutoCAD ADT was developed. The verifying-agents are then available on the server with the fire protection model in order to support the collaboration of all other planners. On the one hand, every planner can retrieve the fire protection model from the server in the form of mobile agents and check his own models for consistency with the fire protection model. The planner is supported by the agent in the form of a check list. A fire protection inspection can occur at the time the agent gets all information needed. On the other hand, the architect can verify the released technical information of the planners for consistency with the fire protection model by mobile verifying-agents. For this purpose he defines the run places for the verifying-agents representing the cooperation partners. The mobile verifying-agent determines the necessary information from every cooperation partner. This can come about automatically, when on the one hand a

link is already defined between the corresponding technical model (e.g. building services model or structural model [8]) and the fire protection model. On the other hand, a communication of the responsible planner can be initiated to link the information.

Each planner makes his model information available to other participants in the design process through a local service-agent. This service-agent has control over the specific model information and can supply the verification. The protection of the planner specific information is guaranteed by encapsulating it within the interface of the service-agent. Other users cannot manipulate the information since the encapsulation is controlled completely by the local service-agent.

5. CONCLUSIONS

This paper described how the technology of software agents enables the integration of different technical models to support collaborative fire protection engineering. A new methodology was developed to verify fire protection models in a distributed environment by means of software agents. This approach integrates the information of building design, structure and technical services with knowledge and rules of German fire protection codes (see Figure 4). The communication between the integrated fire protection model and local technical models of the planners is integrated into the system. The concept allows the integration of fire protection regulations into the planning processes during early design stages. The new software approach enables collaborative engineering with fire protection models and leads to an increase of quality and consistency in the resulting preventive fire protection design system [8] [9].

6. REFERENCES

[1] Rueppel, U.; Meissner, U.: Cooperative Structural Engineering in Distributed Systems. In: Proceedings of the 8th International Conference on Computing in Civil & Building

Engineering, ISBN 0-7844-0513-1, P. 504-509, Stanford University, USA 2000.

[2] Loebbert, A., Pohl, K.D. and Thomas, K.-W.: "Brandschutzplanung fuer Architekten und Ingenieure (Fire Protection Engineering for Architects and Engineers)", Verlag Rudolf Mueller, Koeln, Germany, 2000.

[3] Brenner, W., Zarnekow, R. and Wittig, H.: "Intelligente Softwareagenten (Intelligent Software Agents)", Springer Verlag, Berlin, Germany, 1998.

[4] Finin, T., Fritzson, R., McKay, D. and McEntire, R.: "KQML as an Agent Communication Language", In: Proceedings of the third International Conference on Information and Knowledge Management (CIKM'94), ACM Press, USA 1994.

[5] Labrou, T. and Finin, T.: "A Proposal for a new KQML Specification", In: Technical Report TR CS-97-03, University of Maryland, USA, 1997.

[6] Fipa: "ACL - Agent Communication Language", Foundation for Intelligent Physical Agents (FIPA) specifications homepage. Available online at <http://www.fipa.org>, 06-2002.

[7] Oustershout, J.K.: "Tcl and the Tk Toolkit", Addison Wesley, New York, USA, 1994.

[8] Meissner, U.; Rueppel, U.; Petersen, M.: Cooperative Management of Product Models in the Planning Process of Structural Engineering. In: Proceedings of the International Conference for Computational Engineering Science (ICES2K), ISBN 0-9657001-3-5, P. 240 - 245, Los Angeles, USA 2000.

[9] Rueppel, U.; Theiss, M.: Cooperative Developing of Building Projects in Computer Networks. In: Proceedings of the International Conference for Computational Engineering Science (ICES2K), ISBN 0-9657001-3-5, P. 1182- 1187, Los Angeles, USA 2000.

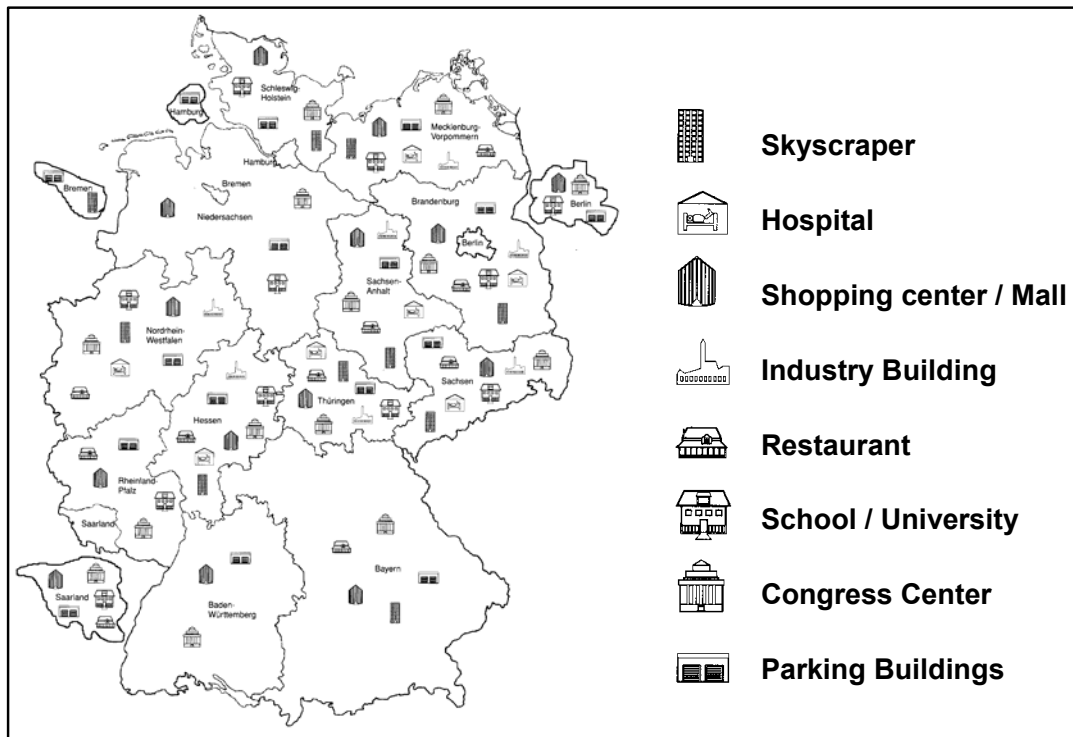


Figure 1: Fire Protection Regulations in Germany

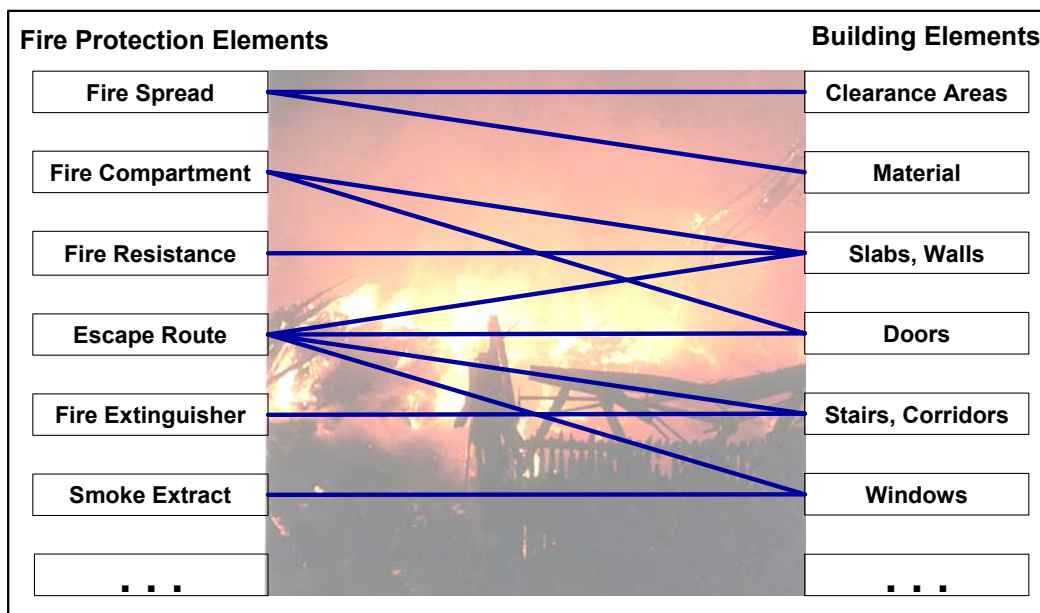


Figure 2: Fire Protection and Building Engineering

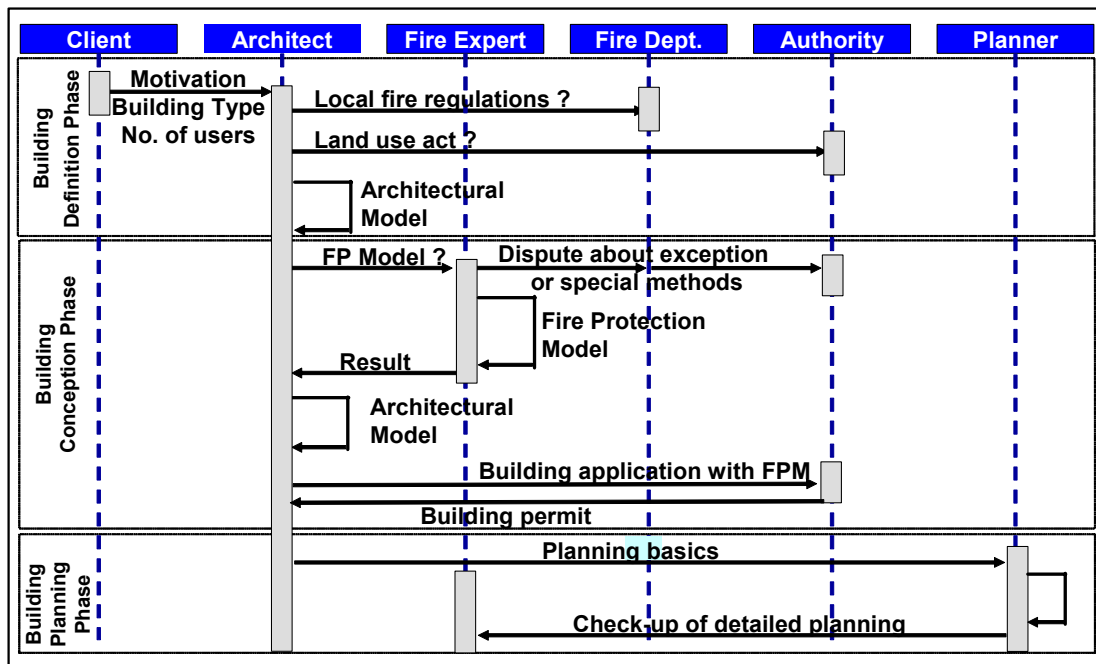


Figure 3: UML Diagram for Fire Protection Planning

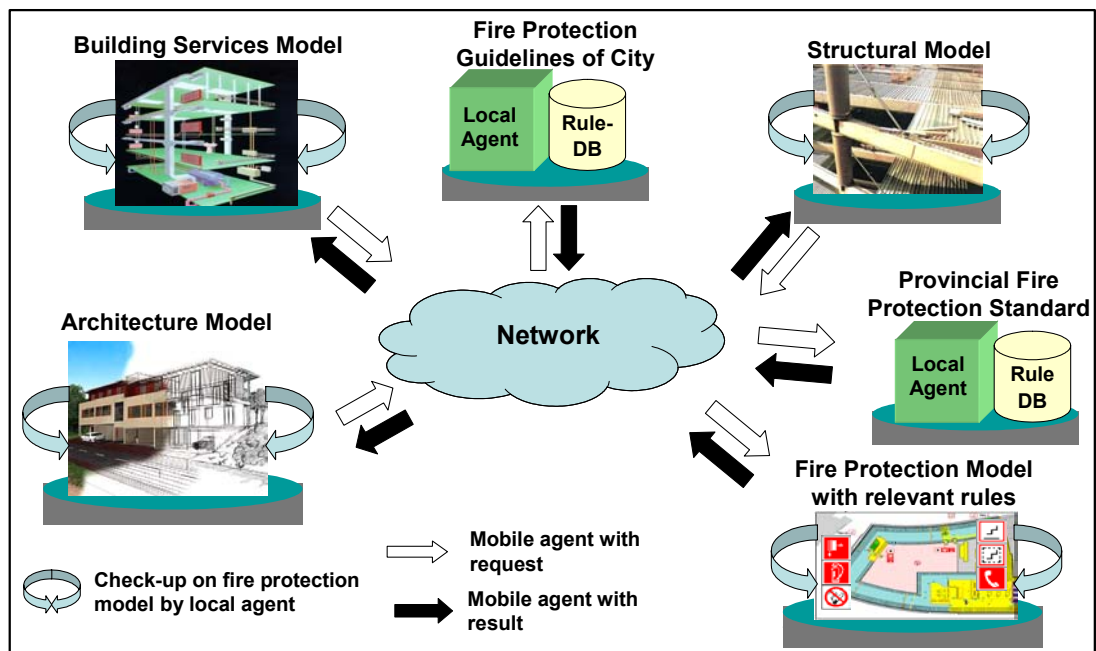


Figure 4: Platform Architecture for Collaborative Fire Protection Engineering