SOFTWARE AGENTS TO SUPPORT DECISION MAKING IN DESIGN AND EXECUTION PLANNING

Hagen Engelmann[1], Fritz Gehbauer[2] and Peter Steffek[3]

ABSTRACT

One of the particular advantages of the Last Planner System is to bring downstream players, for example construction firms, into upstream design and planning phases in order to make optimal use of all available knowledge. This procedure is actually hindered in many countries by the existing governmental contracting rules. To overcome this problem and to be able to include downstream knowledge in earlier phases of the project a solution based on software agents is proposed. The objective is to devise the software architecture for a decision support system incorporating the Last Planner System as communication platform.

As the software agents shall act for persons not available during a Last Planner session they must be integrated in the collaborative process. The paper discusses a concept for this problem based on a Multi-Agent System (MAS) modelling the communication process, and a Service-Oriented Architecture (SOA) to provide different functionalities needed.

The concept provides a corner stone for a computer-based simulation of the knowledge of actors who are not available in phases of the Last Planner process.

KEY WORDS

last planner system, design and execution planning, decision support, multi-agent system, service-oriented architecture

INTRODUCTION

Decision support systems can be designed in various manners including expert systems, simulations and further IT-solutions. The lean construction community has already dealt with the simulation systems for analysis and optimisation in many cases (e.g. Tommelein 1997, Farrar et al. 2004, Srisuwanrat et al. 2007). The paper discusses the possibilities of using multiple software agents within an integrated system of information and simulation components to support the Last Planner System.

The Last Planner System brings downstream players, for example construction firms, into upstream design and planning phases in order to make optimal use of all available knowledge. In early planning phases this procedure is actually hindered in

[1] Research Assistant, Institute for Technology and Management in Construction (TMB), University of Karlsruhe, Am Fasanengarten, D-76128 Karlsruhe, Germany, engelmann@tmb.uni-karlsruhe.de

[2] Professor, Director of the Institute for Technology and Management in Construction (TMB), University of Karlsruhe, Am Fasanengarten, D-76128 Karlsruhe, Germany, gehbauer@tmb.uni-karlsruhe.de

[3] Research Assistant, Institute for Technology and Management in Construction (TMB), University of Karlsruhe, Am Fasanengarten, D-76128 Karlsruhe, Germany, steffek@tmb.uni-karlsruhe.de

many countries by the existing governmental contracting rules applicable in all cases where public companies or public money is involved. If a Last Planner is not able to attend a Last Planner meeting a competent representation is needed. To resolve these problems agent-based software architecture is useful which incorporates the Last Planner System as communication platform. Such a system must be integrated into the communication and cooperation process of the involved persons. Two different types of software agents are proposed to achieve that. The interface agents responsible for the interaction with the human actors and agents supporting Last Planners in their decision or replacing missing human planners. They provide proposals for execution methods and scheduling of activity lists in the different processes of construction.

The interaction between agent and human is a vital point, as the human communication needs to be structured in a way that the agents can understand it and the agents´ suggestions have to be presented in an understandable way. The paper discusses a system architecture based on a Multi-Agent System (MAS) to model the communication process supported by a Service-Oriented Architecture (SOA) providing vital functionalities for the agents.

LAST PLANNER SYSTEM

The Last Planner System (LPS) (Ballard 2000) is a production and time control system designed to build, steer and control a network of commitments, which is required to deliver quality projects in line with the agreed quality, time and costs. The system is based on cooperative interaction, integrates the persons involved in the projects (Last Planners) and ensures that the processes required for a project's completion are understood by the entire project team before the work is started. The Last Planner System triggers a work flow reducing variances and increasing the reliability and thus also the efficiency within the project. Last Planner designates those persons who are responsible for the individual trades (or tasks) as well as for the customers and project managers.

AVAILIBILTY PROBLEM OF THE LAST PLANNERS

The underlying principle of the Last Planner System is to involve the knowledge and experiences of the executing parties in the project as early as possible and to align the project with this production know-how and to consistently optimise it. This production and optimisation-oriented approach to the overall project is implemented by means of intensive communication and agreements between the individual Last Planners. A targeted Look-Ahead-Planning for all necessary works requires a common understanding of the tasks, conditions and existing obstacles and in particular also of the common dependencies involved. A joint knowledge level of everybody involved is necessary to achieve the best possible project results, which can hardly be achieved by the sum of the different suboptimal resolutions of individual works.

In reality it is, however, unfortunate and for several reasons not always possible to use the required Last Planners in the design phases. The attribution of public funds in Germany is subject to the German Construction Contract Procedures (Vergabe- und

Vertragsordnung für Bauleistungen (VOB)), which prevents the involvement of the construction companies in the design and awarding processes. This means that the experiences and know-how of the executing Last Planners are thus not directly available during the design and execution planning. Taking this and other deficiencies into account this research approach attempts to simulate the missing Last Planners through software agents. In doing so, the different steps of the Last Planner must be reflected in the simulation. At the different levels the user can select various interaction scenarios (IS) which take over the following tasks (Figure 1):

- IS1: Interface agent informs about the duration, costs and intermediate deadlines of a previously defined process.
- IS2: Comprises IS1 and the possibility to provide required prerequisites, active solutions and improvement proposals.
- IS3: Depending on the availability of the Last Planner, the software agent in its place enters into the commitments of IS2 and/or several agents simulate the missing Last Planners. This is especially essential in the design phase since the required Last Planners are not available as mentioned above.
- IS4: provides a special interface to monitor and track the commitments.

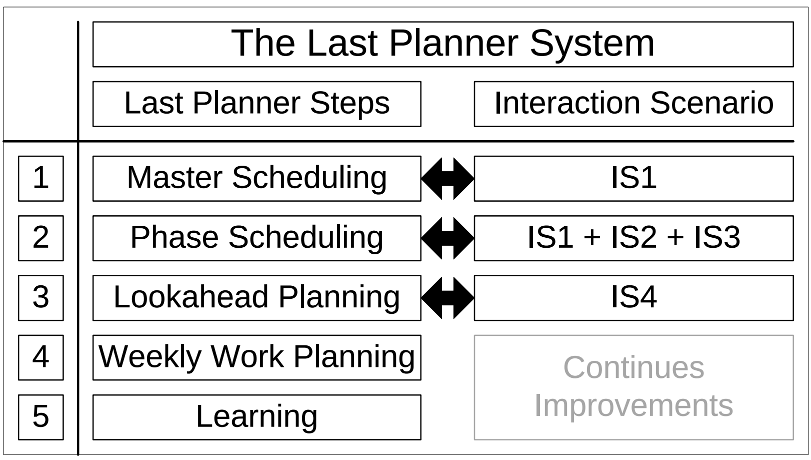

Figure 1: System integration into the Last Planner Steps

Steps 1 to 3 are used in the Last Planner Meetings to consequently look ahead to identify the required processes, existing obstacles and dependencies between the processes. Obstacles are identified and eliminated collaboratively; existing mutual dependencies and/or dependencies between time and place are revealed and taken into account in the Look-Ahead-Plan (step 3). The software agents are used to replace the Last Planners in these steps. In step 1 the software agents can verify the existing deadline forecasts and inform about process lengths of individual tasks. This is reflected by the IS1. In steps 2 and 3 the software agents support the breakdown of the process into individual tasks and display the prerequisites (critical issues). Depending on the Agents' objectives they will present different options and knowledge according to IS1, 2 and 3. To support step 3, the IS4 comprises a review of the commitments made in steps 1 and 2. In step 4 and 5 personal commitments are needed and responses from agents would be counterproductive for the compliance of the Last Planner System. The following sections describe the details of a concept for the realisation of the presented support steps.

AGENT-BASED SYSTEMS FOR DECISION SUPPORT AND SIMULATION

The system concept presented in the following is based on software agents for the interaction with users and decision making components. The agents access Web Services for the simulation of construction processes and additional information about processes and the project.

An agent is a software programme with domain knowledge, goals and actions. Multiple agents interacting in a common environment can build a multi-agent system. A multi-agent system (MAS) is a distributed computing system with autonomous interacting intelligent agents that coordinate their actions to achieve their goal(s) jointly or competitively (Wooldridge 2001). Multi-Agent Systems are used to model a range of problems from Decision Support Systems (DSS) with Artificial Intelligence (AI) to the simulation of complex social and technical systems. There are plenty of examples for the application of agent-based systems (e.g. Sawhney et al. 2003, Mukherjee et. al. 2004) in the field of construction engineering. The authors have successfully implemented a multi-agent system for the simulation and management of earthquake disasters (Engelmann 2006).

Web Services play a key role as enablers of a seamless application-to-application integration both within company boundaries and on a global, cross-organisational scale (W3C 2004). They build the technical foundation for the realisation of Service-Oriented Architectures (SOA) and encapsulate complexity inherent to individual applications and allow their loose coupling. They provide functional building blocks over standard internet protocols independent of platforms and programming languages. In SOA functions are separated in distinct units (services), which can be distributed over a network and can be combined and reused to create business applications. These services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services (Newcomer et al. 2005). Web Services combined with process XML-based modelling languages like the Business Process Modelling Language (BPML) (Arkin 2002) serve as a background for the exchange of information in the proposed system.

DECISION SUPPORT FOR THE LAST PLANNER SYSTEM

The support activity of the agents relies on the ability to simulate construction processes. In construction engineering simulation is a common method in the planning phase as well as for testing new processes and construction methods. A number of simulations for different domains in construction engineering have already been developed. The problem is that no common architecture and no common interface exist for these simulators. In order to get expert knowledge into the design phase of a project, construction engineering companies can provide simulators for their processes, even when they are not allowed to participate with human Last Planners. But companies might not like to share detailed information, so their simulators must be encapsulated as far as possible by only answering specific questions. The access to the internal information has to be restricted.

Web Services build an ideal base for the linking of heterogeneous system. Their major focus is to make functional building blocks accessible over standard internet protocols that are independent from platforms and programming languages. The services can be new applications or just wrapped around existing legacy systems to make them accessible in a network. Construction engineering companies can provide the simulators without the risk of revealing unwanted information as a Web Service only responds to specific requests.

SYSTEM ARCHITECTURE

The system architecture shown in Figure 2 is based on a Multi-Agent System that accesses Web Services. The decision making and decision support tasks are provided by the *Simulation Agent* and the *Last Planner Agent*. Their functions are provided by an *Interface Agent*. The access to information and simulation is provided by Web Services.

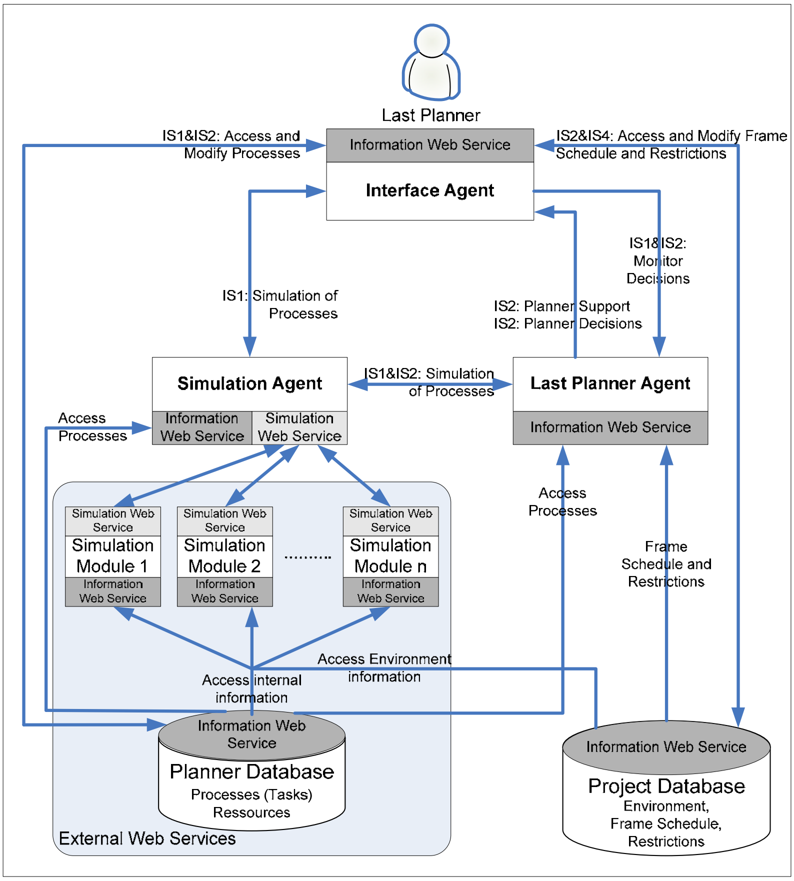

Figure 2: Components of the Last Planner Decision Support System

The *Information Web Services* are wrappers for the database access that provides general information about the project as well as specific data only needed by one Last Planner and his associated decision support and simulation components. The *Project Database* is shared by all planners and includes data about the environment (type of soil, ground water level, construction details, etc.), the current frame schedule and the process plans so far developed by the Last Planner. The *Planner Database* stores the available processes of a specific Last Planner and performance data of machines or personnel needed for simulation. As this may include internal information of a company it should only be available to the simulators and the associated Last Planners. So the Information Web Services of the Last Planner Database have to be restricted.

The *Simulation Web Service* provides a common access method to the simulation modules. A Simulation Agent is responsible for the steering and evaluation of the simulation of a process. A Last Planner *Interface Agent* or Last Planner Agent sends a simulation request consisting of the BPML description of the process. The *Simulation Agent* analyses this process and breaks it down it into tasks. In the simulation context of this paper a task labels a specific process for which a simulation module exists. To break down a process to the task level the *Simulation Agent* accesses the *Planner Database* to obtain the sub-processes needed. Based on this list the different simulation modules are contacted by the *Simulation Web Service*. The result is a time and cost estimation and a list of preconditions needed from other Last Planners during the task execution.

PROCESS SIMULATION

The following section provides a mathematical description to define how the processes are broken down in detail. A generic model of a process p_n can be described by:

$$p_n(\vec{x}_n, \vec{e}) \rightarrow (\vec{y}_n, t\vec{p}_n), n \in \mathrm{N}$$

$\vec{x}$ represents a list of parameters delivered by the user or the previous process, e.g. the start time or the available resources like personnel or machines. $\vec{e}$ includes parameters about the environment of the simulation like the ground water level. The result consists of parameters for the following process in the work flow $\vec{y}_n$ and the preconditions that have to be fulfilled by other processes outside this work flow $t\vec{p}_n$, e.g. deadlines for other planners.

The splitting of the process p_m into a number of n sub-processes and can be describes as following:

$$p_m = p_{mn}\left(p_{m(n-1)}\left(....p_{m3}\left(p_{m2}\left(p_{m1}(\vec{x}, \vec{e}), \vec{e}\right), \vec{e}\right), \vec{e}\right), \vec{e}\right), n \in \mathrm{N}, m \in \mathrm{N}$$

Each sub-process p_{mn} in the work flow depends on the parameters of the previous process. The evaluation starts with the initial process p_{m1}.

$$\begin{pmatrix} p_{mn\,\alpha}\left(\vec{x}_{m(n-1)}, \vec{e}\right) \rightarrow \left(\vec{y}_{n\,\alpha}, t\vec{p}_{n\,\alpha}\right) \\ p_{mn\,\beta}\left(\vec{x}_{m(n-1)}, \vec{e}\right) \rightarrow \left(\vec{y}_{n\,\beta}, t\vec{p}_{n\,\beta}\right) \\ p_{mn\,\chi}\left(\vec{x}_{m(n-1)}, \vec{e}\right) \rightarrow \left(\vec{y}_{n\,\chi}, t\vec{p}_{n\,\chi}\right) \end{pmatrix} \Rightarrow p_{mn}\left(\vec{x}_{m(n-1)}, \vec{e}\right) \rightarrow \left(\vec{y}_n, t\vec{p}_n\right)$$

If a process $p_{m(n+1)}$ has two or more preceding processes executed at the same time (e.g. $p_{mn\,\alpha}$, $p_{mn\,\beta}$, $p_{mn\,\chi}$) a new virtual sub-process p_{mn} is generated which aggregates the results of the parallel processes ($\vec{y}_{n\,\alpha}, \vec{y}_{n\,\beta}$, $\vec{y}_{n\,\chi}$) into one list $\vec{y}_n$ which serves as the input parameter for the following process $p_{m(n+1)}$.

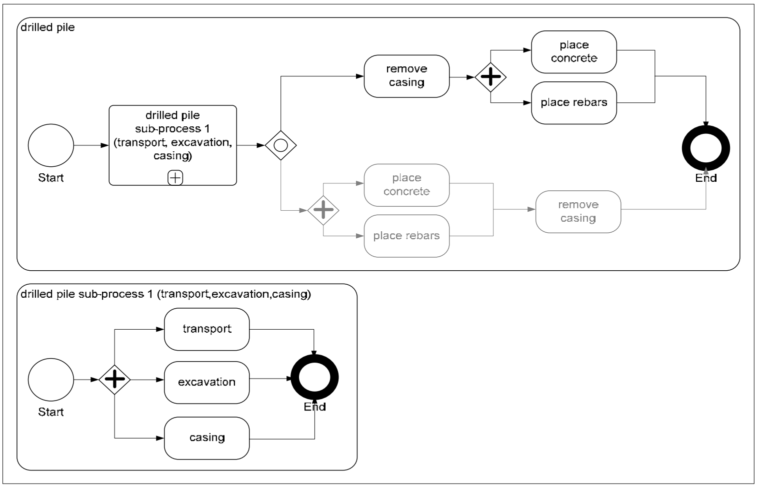

Figure 3: Example for a process model based on the sub-process drilled pile in Business Process Modelling Notation (OMG 2006)

The following example will explain the simulation of the construction process for a drilled pile. As there is the choice of removing the casing before or after the placing of concrete and rebars, the option to remove the casing first is selected for this example. The *Simulation Agent* receives the request to simulate the drilled pile process with given parameters such as the machines used but no further details about the process. The agent starts by breaking up the *drilled pile* process into its four tasks shown in Figure 3. For this it accesses the *Information Web Service* of the *Planner Database* to retrieve the process description in Business Process Modelling Notation. The descriptions of the tasks in the *drilled pile* process include the address to the Web Service of the simulation module for the tasks. As a simulation for the *drilled pile sub-process 1* exists it is handled as a task. The agent executes the tasks in the given order by accessing the corresponding Web Services. The parameters of the call include the ones assigned by the simulation request supplemented by results of previous simulated tasks. Additionally each task simulator accesses the *Project Database* by the *Information Web Service* for general information like the type of soil and the *Planner Database* for internal information needed like the performance data for the machine used for the excavation. The results of each task simulation consist of the parameters for the next tasks and additional information. This includes information for the Last Planner like cost and time estimations as well as preconditions that have to be fulfilled by other processes. These processes can be in the responsibility of other planners, like the levelling of the drilled piles in this example. The Simulation Agent collects all results and prepares a process description in BPML annotated with the results of the simulation.

DECISION SUPPORT AND DECISION MAKING

One of the key elements for the Last Planner System is the communication and cooperation between the actors. A computer assisting the system is not allowed to change this but instead needs to be adapted and integrated. Even then the acceptance and trust towards the computer support must evolve. Additionally, to be able to act in behalf of a missing human actor the Last Planner Agent has to adapt to the preferences of its Last Planner. Taking this into account the computer support in this concept includes four interaction scenarios already mentioned in Figure 1. The following will describe the realisation of the scenarios shown in Figure 4.

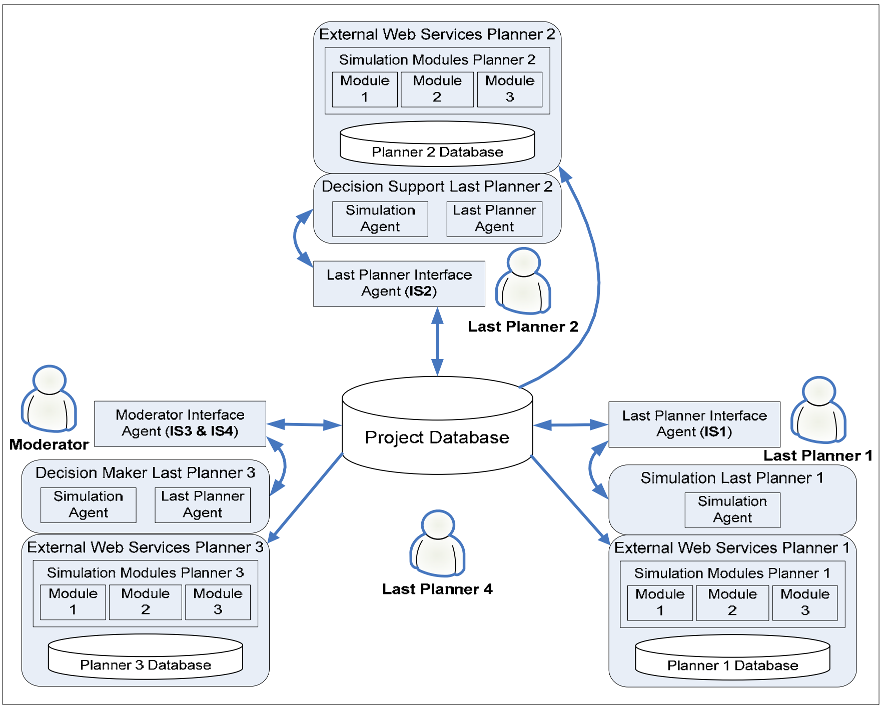

Figure 4: Usage of decision support components in a Last Planner meeting

In ***IS1*** the *Interface Agent* will only present the GUI elements for the configuration of the process simulation. The interface will include elements for the modelling of new processes and sub-processes based on the available tasks and already existing processes. These are retrieved from the *Information Service* of the corresponding *Planner Database* which acts as a repository for all available process descriptions. To simulate a process the *Interface Agent* sends a simulation request to the *Simulation Agent*, which send the results as an annotated BPML diagram. The results of the simulations support the user in the evaluation of the best matching process for the given situation.

IS2 provides decision support for the Last Planner. Based on the processes available in the *Planner*

Database the *Last Planner Agent* gives advice for the selection of a process. For this the agent simulates the available solutions under the given restrictions of the current situation, e.g. starting time, end time, costs and promises for deadlines of other Last Planners. As a result the *Interface Agent* presents the simulated processes. As advice preferable processes are marked and proposals are given on how changes to the deadlines of other planners can improve the solution. The user decisions are monitored by the *Last Planner Agent* to learn the user preferences. Users interacting in IS2 will most likely be persons that are allowed to access internal information about the company processes. Because of that they should have full access to the *Information Web Service* of the *Planner Database*, while in all other interaction scenarios the user access is limited.

To act in behalf of a missing Last Planner in the **IS3** the agent needs a sufficient repository of available processes. To make decisions based on the style of a specific person a sufficient knowledge about their personal preferences is also needed. A *Last Planner Agent* can act for one missing planner and several *Last Planner Agents* can negotiate in a virtual Last Planner Meeting:

- In the case of a missing person the moderator can access the deadlines proposed by the agent via the Interface Agent. This will also include statements on possible improvements by changes in the proposals of human planners. Compared to IS2 the agent only presents one solution, which changes if the moderator modifies the deadlines for the agent processes.
- In the case of different *Last Planner Agents* independently negotiating a process schedule a human-like communication should be established. To reproduce this communication the FIPA agent specification (2006) provides communicative acts that can model a natural conversation. As before each agent reacts based on its knowledge about processes and the preferences of his owner. This mode demands simulators for all task needed in a construction process and detailed information about all sub-processes.

The ***IS4*** supports the look-ahead-planning in the Last Planner System. If the process deadlines of all Last Planners are stored in the Project Database, it is easy to compare the target state and the actual state as well as identifying variances and their roots.

The idea to the approach described here is based on research dealing with the support for decision makers of disaster response operations after earthquakes. The result of this work is the so-called Disaster Management Tool (DMT) (Gehbauer et al. 2007), a distributed system which has already been tested by crisis management organisations in Rumania and Germany. It shows the potential benefit of the presented ideas. During the response to a disaster decision makers have to deal with insufficient resources in relation to the damages. The DMT provides different support option to assist them in allocating the optimal resources. The system evaluates user decisions by simulating resulting damages. Agents give active advice for the resource allocation and warn decision makers of potential problems, e.g. when allocated resources are insufficient or inappropriate for a task. To evaluate the preparedness of an organisation the decision support agents can act independently to develop solutions for disaster scenarios.

CONCLUSION AND PERSPECTIVE

This paper presented a concept for the support of the Last Planner by a computer system. A realisation of all features presented here seems impossible at the moment because of missing simulators for all construction processes. But their already exist ambitions for an easier integration of simulation systems in the construction processes (Chahrour et al. 2006). Also step 5 of the Last Planner System presents an interesting opportunity to collect knowledge needed for the simulation of new processes or the improvement of existing simulators. How this could be done in a more or less automated way is an initial point for further research.

Nevertheless a common Web Service interface would be a major step forward bringing scientific work to practical use. For companies such a standard interface offers the chance to integrate their knowledge early in the planning process without the risk to divulge too much of their internal information. This may be one of the main advantages of the concept and a key enabler for its implementation.

REFERENCES

Arkin, A. (2002). "Business Process Modeling Language (BPML)." BPMI.Org, (available at http://xml.coverpages.org/BPML-2002.pdf).

Ballard PhD (2000). *The Last Planner™ System of Production Control*. Dissertation, School of Civil Engineering, The University of Birmingham, U.K.

Chahrour R., Franz V. (2006). Seamless data model for a CAD-based simulation system. *Proccedings of the Joint international conference on computing and decision making in civil and building engineering- Building on IT*

Engelmann, H., F. Fiedrich, et al. (2006). An Architecture for Decision Support and Training in Emergency Operation Centres. *100 th Anniversary Earthquake Conference*.

Gehbauer, F., Markus, M., Engelmann, H., Popa, I., Schweier, C., Rehor, M., Werder, S. (2007): The Disaster Management Tool (DMT). *Proceedings of the International Symposium on Strong Vrancea Earthquakes and Risk Mitigation*, Oct. 4-6, 2007, Bucharest, Romania, pp. 421 – 431.

Farrar, J. M., S. M. AbouRizk, et al. (2004). "Generic Implementation of Lean Concepts in Simulation Models.". *Lean Construction Journal 1*.

FIPA (2006). *FIPA Communicative Act Library Specification.* (available at http://www.fipa.org/specs/fipa00037/SC00037J.pdf).

Mukherjee, A., E. M. Rojas, et al. (2004). Implementing a General Purpose Framework using Multi-Agents for Construction Management Education. *2004 Winter Simulation Conference*.

Newcomer, E. and G. Lomow (2005). *Understanding SOA with web services*. Upper Saddle River, NJ, Addison-Wesley.

OMG (2006). *Business Process Modeling Notation Specification,* (available at http://www.omg.org/docs/dtc/06-02-01.pdf).

Sawhney, A., H. Bashford, et al. (2003). Agent-Based Modeling and Simulation in Construction. *2003 Winter Simulation Conference*.

Srisuwanrat, C. and P. G. Ioannou (2007). The Investiagation of Lead-Time Buffering Uunder Uncertainty using Simulation and Cost Optimization. *Proceedings of the 15 th International Group of Lean Construction.*

Tommelein, I. D. (1997). Discrete-event Simulation of Lean Construction Processes. *Proceedings of the 5 th International Group of Lean Construction.*

W3C (2004). Web Services Architecture, *W3C Working Group Note 11* (available at http://www.w3.org/TR/ws-arch/wsa.pdf).

Woolridge, M. and M. J. Wooldridge (2001). *Introduction to Multiagent Systems*, John Wiley & Sons, Inc. New York, NY, USA.