A PROCESS-ORIENTED COOPERATION MODEL FOR DISTRIBUTED CIVIL ENGINEERING CONSTRUCTION WORKS

Rolf Katzenbach [1], Uwe Rüppel [2], Udo F. Meißner [2], Johannes Giere[1], Armin Wagenknecht[2]

ABSTRACT

The planning processes in Civil Engineering have specific aspects compared to the design-, construction- and maintenance processes in other industry domains: In opposition to industrial production processes, Civil Engineering always deals with unique structures. Planning is characterized by a high degree of specialization and a great division of work. Planning participants operate in legally independent and temporarily and physically distributed organizations. They have to co-operate and communicate in a heterogeneous planning environment. In order to develop appropriate methods and tools for the process modeling in Civil Engineering, the specialization and the distribution are important aspects that have to be considered. This paper presents a new approach to better coordinate the planning processes based on a generic process model, which uses semantic information within the theoretical background of Coloured Petri Nets. Planning information is represented in metainformation and is linked to the process model, so that it can be accessed to support decisions. The approach uses process model patterns which are predefined for each construction element. They match a general formalism and are compatible to each other. The process model can thus be adapted to the proceeding planning process and meets the requirements of the dynamic change. The contribution gives an overview on the approach to handle the distributed process modeling in Civil Engineering for cooperation purposes and shows the pilot-implementation.

KEY WORDS

Process-Modelling, Petri Nets, Net-based Cooperation.

[1] Institut für Geotechnik, Technische Universität Darmstadt, Petersenstraße 13, D-64287 Darmstadt, Germany, Phone +49-6151-162049, FAX +49-6151-166683, sekretariat@geotechnik.tu-darmstadt.de

[2] Institut für Numerische Methoden und Informatik im Bauwesen, Technische Universität Darmstadt, Petersenstraße 13, D-64287 Darmstadt, Germany, Phone +49-6151-163444, FAX +49-6151-165552, sekretariat@iib.tu-darmstadt.de

1 INTRODUCTION

In Civil Engineering projects architects, engineers, authorities and craftsmen create complex buildings with a unique design. Thereby, the planning participants always have to incorporate current scientific know-how and technical innovations in their planning and design activities in order to create a building with technical perfection with respect to economical and ecological aspects. Therefore, the planning processes in Structural Engineering are characterized by a high degree of specialization and a great division of work. Consequently, a lot of engineers, who operate in legally independent as well as temporarily and physically distributed organizations, have to co-operate and communicate in a heterogeneous planning environment.

Figure 1 illustrates the approach for the network-based co-operation platform, the so-called Co-operation Model Geotechnics, consisting of the following three essential elements:

- Local Process Domains
- Information Packages
- Global Process Management System

Within the Co-operation Model Geotechnics each engineering organization is represented by a Local Process Domain. Providing the necessary resources of hardware, software or communication each Local Process Domain supports the corresponding organization and ensures its integration into the communication network. The paper focuses on the Local Process Domain Geotechnics to model the processes in geotechnical design and construction and to tie the Geotechnical Engineer into the Co-operation Model [Katzenbach et al. 2002].

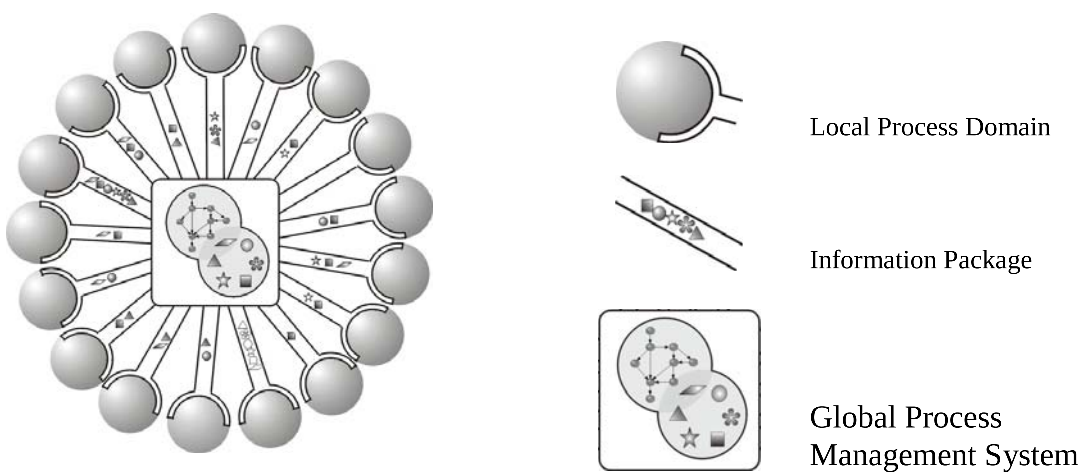


Figure 1: Co-operation Model Geotechnics consisting of Local Process Domains, Information Packages and the Global Process Management System

Essential parts of the Co-operation Model are the Information Packages, represented by the symbols within the information channels of Figure 1. The exchange of information between different planners is organized in Information Packages which consist of two parts:

elementary information units describing distinct aspects of the project, e.g. geometrical, structural or technical information and metainformation providing the necessary information for the control of processes.

The common co-operation platform for all processes is the Global Process Management System. Based on the process model, the process simulation and by evaluating the metainformation the Global Process Management System dynamically activates communication channels between different engineering organizations.

In the following Chapters the different components of the Global Process Management System are presented. Beginning with a suitable process model (Chapter 2) an approach which provides modeling on different detail-levels will be described (Chapter 3). In order to enrich the semantical power of the Information Packages, metainformation structures are used (Chapter 4). Concerning the reusability of modeled processes and the thereby generated modeling-experience, the usage of design patterns is discussed in Chapter 5. Finally, the pilot-implementation for the Global Process Management System is introduced (Chapter 6).

2 SUITABLE PROCESS MODEL: PETRI NETS

Petri Nets provide both a mathematical formalism and a graphical representation based on the graph theory in order to model the concurrent and asynchronous behavior of a discrete system [Petri 1962]. Since the invention of Petri Nets, various researches, extensions and improvements have been applied to the original Petri Net theory. Application of Petri Nets to process modelling and workflow management has been introduced by, e.g., v.d. Aalst [Aalst 1998] and Oberweis [Oberweis 1996]. Especially, the application of Petri Nets on Civil Engineering processes is explained in [Rueppel et al. 2003]. The main reasons for using Petri Nets as Process Model of the Global Process Management of the Co-operation Model Geotechnics are:

- Petri Nets have consistent formal semantics, i.e., a workflow modelled as a Petri Net has a clearly defined meaning without leaving any interpretation possibilities to the modeller.
- Petri Nets have a graphical representation. Thus, workflow models using Petri Nets are quite intuitive.
- Petri Nets provide all basic concepts for modelling typical workflow constructs like sequence, parallelism, iteration or synchronisation. Furthermore, Petri Nets explicitly enable the modelling of states within a workflow.
- Petri Nets enable the modelling of dynamic aspects, i.e., by use of so-called tokens the flow of arbitrary objects processed within a workflow can be modelled.
- Petri Nets provide various analysis possibilities. Petri Nets are bipartite directed graphs extended by the token concept. Thus there exist various structural and behavioural possibilities of representation.

For a short introduction to Petri Nets see [Aalst 1998], for a comprehensive introduction [Reisig 1986] or [Baumgarten 1990] are recommend. As illustrated in Figure 2 a Petri Net consists of places, transitions and arcs, with each arc connecting either a transition and a place or a place and a transition. The tokens reside on the places. Based on well defined rules the transitions can "fire" and thus let the tokens "flow" through the net.

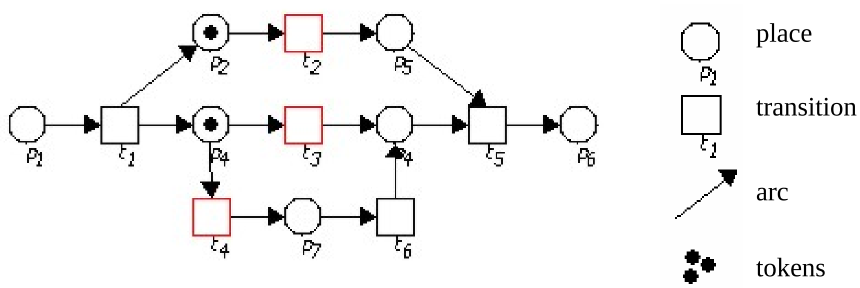


Figure 2: Petri Net

The basic idea in modeling Civil Engineering processes with Petri Nets is to describe planning states with places, planning activities with transitions, planning dependencies with arcs and planning information with tokens. Places, transitions and arcs form finite sets. Furthermore, Petri Nets allow to model processes on different hierarchy- and detail-levels. The following Chapter describes the advantages of this technique for the process modeling in Civil Engineering [Rueppel et al. 2004].

3 DISTRIBUTED PROCESS MODELLING

The distributed process modeling is based on hierarchical Petri Nets in the approach of [Jensen 1997] with the substitution of transitions based on distinct places so-called "socket places" and "port places". In [Aalst 1999] another approach to manage interorganizational workflows based on Petri Nets is presented. The approach focuses on loosely coupled inter-operability between business or planning partners. Anyhow, the defined Petri Net sets for modeling asynchronous and synchronous communication between planning participants defined in [Aalst 1999] can extend the approach introduced in the following [Berkhan et. al 2005].

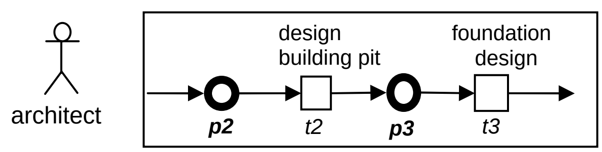


Figure 3: Architect Modeling a Coarse Process Model with Socket Places (places p_2 and p_3)

The central and coarse process model provides only an abstract description of the whole planning process. This model is generated by the architect or a project manager, respectively. A transition, i.e., a planning activity, which will be described in more detail by a distributed process model, is identified by socket places as input- and output places. Figure 3 shows an architect modeling a coarse process net – from his point of view – just as a simple sequential process. The place p2 denotes an input-socket place, the place p3 denotes an output-socket place and the enclosed transition t2 indicates a coarse description of a planning activity, e.g. "designing building pit".

On the other hand organizational distributed engineers generate their workflow nets describing technical aspects of their planning activities in detail. Essentially, these nets have to fulfill the requirements of a Workflow Net defined in [Aalst 1998] with a single input place (p1) and a single output place (p2). Input place and output place are denoted as "port places".

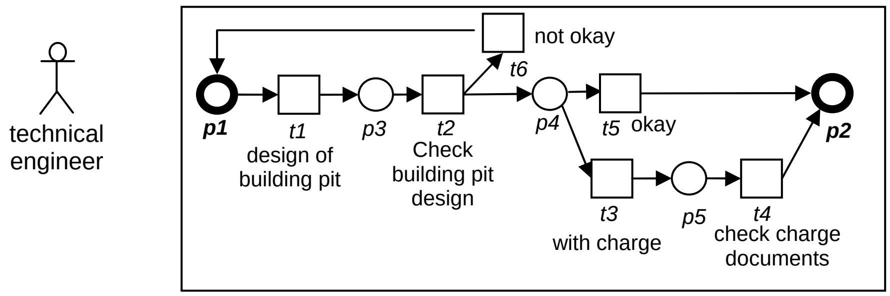


Figure 4: Technical Engineer Modelling a Detail Process Model with Port Places

The "technical engineer" in Figure 4 e.g. represents a geotechnical engineer who models the planning activities and planning states in order to design a building pit more detailed.

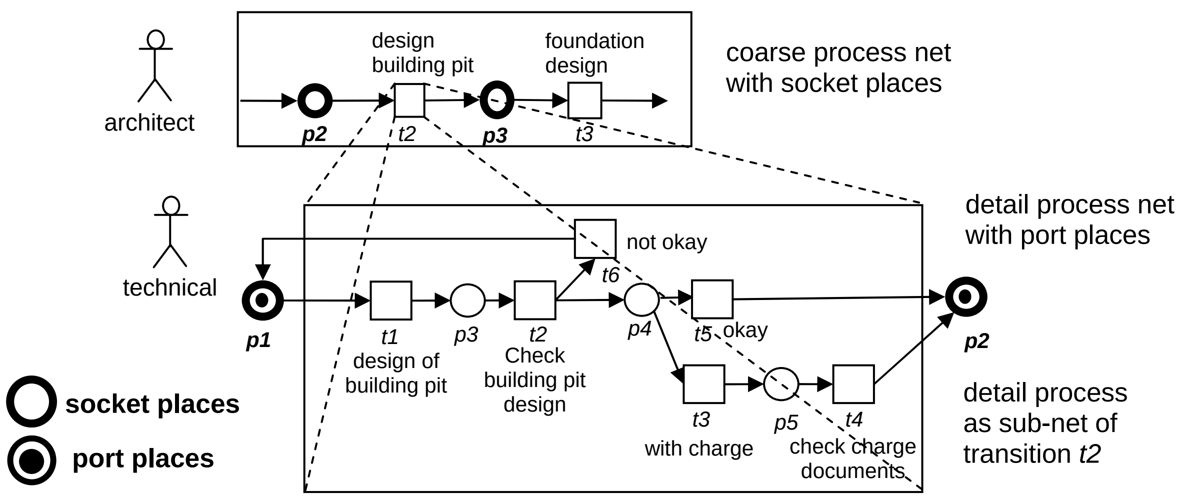


Figure 5: Detailed Process Net as Subnet of Transition t_3 in the Coarse Process Net

The combination of the coarse process net and the distributed detail process nets is based on the unique assignment of port places to socket places. For this assignment the introduced software ProMiSE (Chapter 6) supports a manual assignment. Once the places are assigned, the detailed process net is incorporated in the coarse process net as subnet of the denoted transition t2 (Figure 5).

Besides the processes, which can be modeled distributed with hierarchical Petri Nets, it is essential to integrate geometrical, structural or technical information in the process model. In order to use this information to control the modeled processes, the implementation of metainformation is necessary, which is described in the next Chapter.

4 METAINFORMATION

Generally speaking, planning is the process of generating, exchanging and processing information. Up to date information exchange in Civil Engineering is mainly document-based: plans, calculations and text-documents are the main document-types being exchanged. In the reference project of the 38-storey high-rise office-building Gallileo in Frankfurt/Main the log-data of the project-server could be analysed. The project-server was used during the whole construction phase for all domains. All planning participants were obliged to use the project server for information exchange. The analysis of the log-data showed the following characteristics:

Table 1: Analysis of the exchanged documents in the reference project

Type	Amount
plan (dwg / plt)	90,8 %
spreadsheet (xls)	5,0 %
pdf	3,2 %
other	1,0 %

Total upload	61676 docs
Total download	93788 docs

Consistent electronic product-models only exist so-far for specific domains, such as steel-construction. With reference to the process modeling it is essential to extract the process relevant information from the abundance of all information being exchanged. Information is considered process relevant, if it is needed in decisions, where – depending on the content of the planning – two or more different tasks are to be carried out alternatively. As it is derived from the actual information and describes the planning content, this information is called metainformation. It may be derived from different sources. Of course, the difficulty lies in the definition of the adequate amount of information. One approach from the technical basis is to abstract the information from technical standards and regulations [Katzenbach et al. 2004].

As for the process model, the metainformation has to be accessible during run-time. In the Co-operation-model the Standard Meta Language (SML, [Milner et al. 1997]) was chosen as it gives good support for accessing, evaluating and modifying the information in the model.

As a first approach tuples of labels (identifiers), values and indexes were used. The tuple includes construction specific parts of information and can be addressed by the label. All information is gathered in a list as an Information Container. Both, the tuples and the list have an index, allowing it to distinguish versions of information and different information containers (Figure 6).

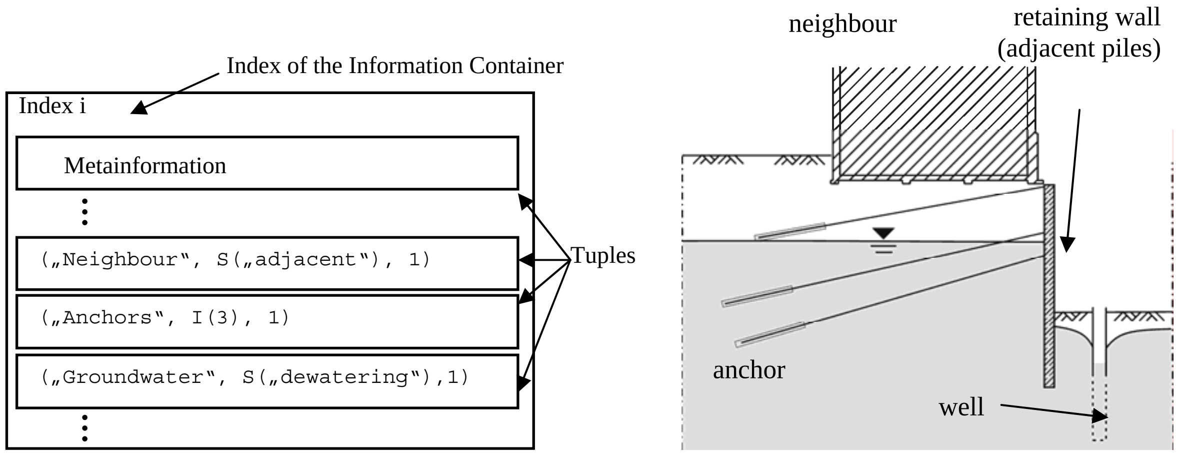
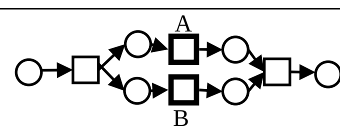
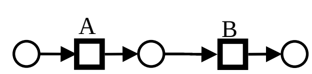
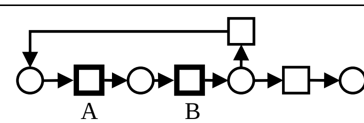


Figure 6: Information Container in SML (Sample)

5 DESIGN PATTERNS

With respect to the necessity to adapt the process model during the planning process, a modular approach with design patterns seems suitable. The aim of the use of design patterns is to build up the whole complex process model from small parts in a bottom-up approach. As the expertise to define these processes and to set up process models as design patterns is linked to the professional expertise in every domain, the modular approach enables distributed preparation of the patterns. On the long run process knowledge will thus become part of the expertise and will be accessible in the process patterns, similar to technical knowledge in standards and regulations.

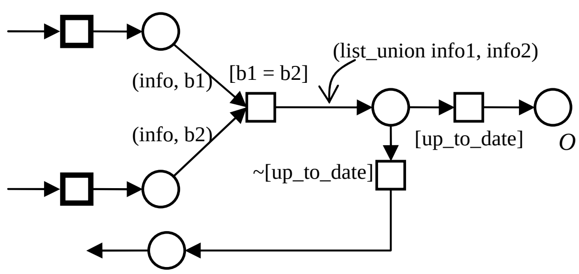
Table 2: Design patterns in Petri Nets

dependency	**primitives**	**Design patterns in Petri Nets**
no dependency	concurrency	
unilateral dependency	sequence	
mutual dependency	sequence + iteration	

Of course there are formal requirements to the design patterns. The basic approach focuses on the possible dependencies between two activities A and B. Six routing primitives (Sequence, Iteration, XOR-Split, XOR-Join, AND-Split, AND-Join) are arranged into three elementary design patterns, taking into consideration the kind of dependency between two activities in the design process (Table 2).

For the mutual dependency of activities A and B it is necessary to carry out both activities until the result of the activity B has no effect on A. An example for such a mutual dependency is the design of a combined piled-raft foundation, where the load-settlement behaviour of the whole foundation interacts with the rising structure. Structural engineer and Geotechnical Engineer have to cooperate iteratively to optimize the construction. Both planning domains have to approve the design before the next steps can be taken. In the design pattern this is symbolized by the XOR-split following activity B.

The primitives in Table 2 can be extended by incorporating the metainformation e.g. at decisions points. Thus the process is driven by the actual planning results. The Petri Net Syntax has then to be extended with colors to be able to embody information in the current marking. For this purpose a pilot implementation using SML was set up. Figure 7 gives an example of a synchronization and decision detail.




```
color Int_Real_String = union I:Int + R:Real + S:String;
color Meta = Product String * Int_Real_String * Int;
color Info = list Meta;
...
var info, info1, info2: Info;
var b, b1, b2: Int;
...
(* list_union unifies two lists of Metainformation *)
fun list_union ([], list2) = list2

|list_union (hd::tl, list2) = exists_l hd (list_union(tl,list2)) ;
```

Figure 7: Detail of process model

All necessary functions and tools were installed in an SML-Library, so that the necessary operations on the metainformation in the information container (represented as tokens) could be performed. Splitting, unifying and versioning of the information were implemented.

A further formal requirement to the design pattern is a process model in accordance with the definitions of v.d. Aalst given for the Workflow-Nets, which have already been mentioned in Chapter 3. Both, the design patterns – including the function library to operate on the metainformation – combined with Workflow-Nets are an universal approach to design processes. Functionality and universal use of the model is sufficient, in spite of the coarse information model used.

The aspects, explained in the previous Chapters, are fully realized in the reference implementation ProMiSE which will be introduced in the coming final Chapter.

6 REFERENCE IMPLEMENTATION PROMISE

For the management of engineering processes in heterogeneous computer networks the network enabled software tool ProMiSE was developed at the Institute of Numerical Methods and Informatics in Civil Engineering in close co-operation with the Institute of Geotechnics at the Darmstadt University of Technology [Rueppel et al. 2005]. ProMiSE is based on Petri Nets with individual tokens to represent and evaluate process relevant product model information. ProMiSE is used for

- process design, definition and analysis at build time, and
- process instantiation, control and user/software interaction at run time.

From the developer's point of view the implementation of ProMiSE is based on two concepts:

- firstly, object-oriented programming approaches have been implemented in order to realize graph theoretical algorithms, workflow specific analysis properties – soundness property – and network communication mechanisms.
- secondly, existing Java Petri Net archives, like the "Platform Independent Petri Net Editor" (PIPE), have been integrated. PIPE provides basic Petri Net analysis possibilities, e.g., liveness, boundedness or safeness [Bloom 2003].

In ProMiSE the information representation is realized with the Petri Net Markup Language (PNML) [Kindler 2002]. The PNML format is the basis for the file-based input/output, the storage in XML databases or the network exchange based on WebServices. Especially, the ProMiSE interface to PIPE uses the PNML information representation with the JDOM API [Hunter/McLaughlin 2000].

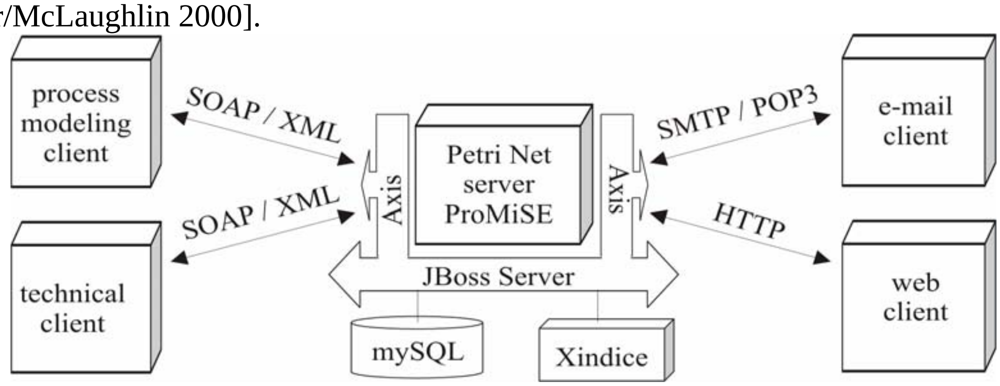


Figure 8: System Architecture of ProMISE

The Petri Net based process control software tool ProMiSE is embedded in the JBoss application server with interfaces to the relational database system MySQL and the native XML database Xindice [Xindice 2001]. The network interaction between ProMiSE and client applications during process run time is realized with WebServices based on the Apache Axis implementation [Axis 2002] [Chappell/Jewell]. Figure 8 illustrates the ProMiSE architecture comprising the ProMiSE server and different client applications like standard internet applications, e.g., an E-mail-client, a process modeling client and technical applications, e.g., GAPP ("Geotechnical Application for Product and Process Modeling).

ACKNOWLEDGMENTS

We would thank the German Research Foundation for financing this project which is part of Research Priority Program 1103 "Network-based Co-operative Planning Processes in Civil Engineering" of the German Research Foundation (DFG).

REFERENCES

v.d.Aalst, W. (1998). "The Application of Petri Nets to Workflow Management." *Journal of Circuits, Systems, and Computers* 8 (1) 21-66, World Scientific Publishing Company, Singapore u.a.

v.d.Aalst, W. (1999). "Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets". *Systems Analysis-Modelling-Simulation*, 34 (3) 335-367.

Axis 2002. *Apache Axis*, The Apache Project, http://ws.apache.org/axis.

Baumgarten, B. (1990). *Petri-Netze – Grundlagen und Anwendungen*, Spektrum-Akademischer-Verlag.

Berkhahn, V, Klinger, A., Rüppel, U., Meißner, U.; Greb, S., Wagenknecht, A. (2005). "Process Modelling in Civil Engineering based on Hierarchical Petri Nets." Scherer, R. J.; Katranuschkov, P.; Schapke, S.-E. (Ed.) *22 nd Conference on Information Technology in Construction (W78*), 304 (1) 51-56, Dresden.

Bloom, J. (2003). *Platform Independent Petri Net Editor (PIPE)*, Imperial College, London, Computer Department, http://petri-net.sourceforge.net.

Chappell, D.A., Jewell, T. (2002). *Java Web Services*, O'Reilly.

Hunter, J. & McLaughlin, B. (2000). jdom.org, http://www.jdom.org.

Jensen, K. (1997). "*Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*". Volume 3, Practical Use. Springer-Verlag, Heidelberg.

Katzenbach, R., Meißner, U. F., Rüppel, U., Giere, J., Greb (2002). "Process-Oriented Networked-based Collaboration in Geotechnical Engineering." *Proc. of the 9th International Conference on Computing in Civil & Building Engineering*, 1063-1068, Taipei, Taiwan.

Katzenbach, R.; Giere, J. (2004a). "Coordinating Planning Processes in AEC using an Adaptable Process Model." *Proc. of the Xth International Conference on Computing in Civil and Building Engineering*, Weimar, Germany.

Katzenbach, R.; Meissner, U. F.; Rueppel, U.; Savidis, S.; Giere, J.; Greb, S. und Mejstrik, M. (2004b). "Abstraction of Process Relevant Information from Geotechnical Standards and Regulations." *Proc. of the Xth International Conference on Computing in Civil and Building Engineering*, Weimar, Germany

Kindler, E. (2002), "The Petri Net Markup Language." H. Weber (Ed.), *Petri Net Technology for Communication Based Systems*, Springer LNCS.

Milner, R.; Tofte, M., Harper, R. and MacQueen, D. (1997). "*The Definition of Standard ML*". The MIT Press, Cambridge, USA.

Oberweis, A. (1996). "*Modellierung und Ausführung von Workflows mit Petri-Netzen*". Teubner, Stuttgart; Leipzig.

Petri, C. A. (1962). "*Kommunikation mit Automaten*". (Diss.), Schriften des IIM 3, Universität Bonn, Institut für Instrumentelle Mathematik, Bonn, Germany.

Rueppel, U., Greb, S., Theiss, M. (2003). "Managing Distributed Planning Processes in Fire Protection Engineering based on Agent Technologies and Petri-Nets." *Proc. of the 10th ISPE International Conference on Concurrent Engineering*, Book 2, 651-656, Portugal.

Rueppel, U., Meißner, U. F., Greb, S. (2004). "A Petri Net based Method for Distributed Process Modelling in Structural Engineering." *Proc. of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)*, Weimar.

Rueppel, U.; Meißner, U. F.; Greb, S. (2005). "Process-orientation based on Petri-Nets for the Coordination of Planning Processes in Civil Engineering" *Proc. of the International Conference on Computing in Civil Engineering 2005 (ICCC2005)*, Cancun, Mexico.

Reisig, W. (1986). "*Petrinetze - Eine Einführung*". 2. Edition, 1-196, Springer, Berlin.

Xindice (2001). *Apache Native XML Database*, The Apache Software Foundation, http://xml.apache.org/xindice.