Product Models in Network Based Co-operation in Structural Engineering

B. Firmenich, Bauhaus University Weimar, 99421 Weimar, Germany
(berthold.firmenich@bauing.uni-weimar.de)

Summary

The Priority Programme 'Network Based Co-operation in Structural Engineering' of the 'German Research Foundation' (DFG) has been established in the year 2000. This paper describes and discusses the main research directions and first results of the workgroup 'Distributed Product Models'.

The five projects of the workgroup have developed completely different solutions for specific application domains. Each solution concept deals with a consistent product modeling and knowledge processing in a distributed environment in the planning process. The individual solution approaches of the projects are described and the underlying basic assumptions are discussed. A unified system architecture is described for all projects of the workgroup. Two different approaches (object-oriented and graph-based models) have been introduced for product and knowledge modeling. The common structure of these models will be explained to fully understand the differences of these modeling approaches. Finally the concepts for co-operative work and conflict management in a distributed environment are described: The solution approaches will be distinguished by classifying the supported co-operation according to time.

A final scientific summary describes the state-of-the-art in network based co-operation in structural engineering: The role of research directions like knowledge modeling, standard product modeling and versioning in the distributed planning process will be explained.

1 Introduction

DFG: The Deutsche Forschungsgemeinschaft (German Research Foundation) is the central, self-governing research organisation that promotes research at universities and other publicly financed research institutions in Germany. A particular funding instrument of the DFG are the Priority Programmes: The purpose is to advance currently relevant fields in science and the humanities by encouraging coordinated, interdisciplinary, national and international co-operation between outstanding researchers (DFG 2004).

Priority Programme SPP 1103: In the field of Computer Aided Civil Engineering in Germany (Bauinformatik) the Priority Programme 'Network Based Co-operation in Structural Engineering' (SPP 1103) has been established in the year 2000 (Meißner and Rüppel 2004). The funding duration is, as a rule, limited to six years. As many as 14 projects of different researchers are currently participating in the Programme. So far, a wide range of solution approaches has been proposed.

Workgoup 'Distributed Product Models': To promote project spanning research results each participating project has been assigned to a workgroup inside the Priority Programme. This paper describes the main research and first scientific results in the workgroup 'Distributed Product Models'. The author is the speaker of this workgroup since 2003.

2 Main research directions

Participating Projects: The workgroup comprises five projects managed by professors from German universities: Professor Beucke (Bauhaus University Weimar), Professor Nagl (Aachen University of Technology), Professor Pegels (Wuppertal University), Professor Scherer (Dresden University of Technology) and Professor Schnellenbach-Held (Essen University). For practical reasons the projects will be identified in this paper by the university towns.

Co-operation matrix: Naturally, the main research objectives of the participating projects are very diverse. To illustrate the main topics of each project, the coordinators of the Priority Programme have designed the so-called co-operation matrix with projects in rows and main research objectives in columns (Figure 1). Research topics addressed by the projects are marked by an 'x' in the respective matrix element. Figure 1 shows the part of the matrix relevant for the workgroup ‘Distributed Product Models’.

	Distributed Planning Processes				Co-operation Concepts			Integration Concepts			Knowledge Processing	
	Distributed Models	Distributed Simulation	Mobile Processing Methods	Process Description and Simulation	Model Consistency	Agent Based Communication	Process Control Semantics	Middleware	Modeling and Exchange Specifications	Agent Wrapping	Knowledge Bases	Agent Based Knowledge Processing
Prof. Beucke	x				x							
Prof. Nagl	x				x			x	x		x	
Prof. Pegels	x				x				x			
Prof. Scherer	x				x			x				
Prof. Schnellenbach-Held	x				x			x			x	

Figure 1: Co-operation matrix for the workgroup according to (Meißner and Rüppel 2004)

The co-operation matrix shows the main research directions of the participating projects of the workgroup ‘Distributed Project Models’: The research objectives ‘Distributed Models’ and ‘Model Consistency’ can both be addressed in a distributed environment with a logical shared product model. Product models can only be kept consistent if technical know-how and knowledge about valid standards is available. Due to the complexity of today’s building projects it becomes more and more difficult to define all the necessary knowledge at compile-time. Therefore, two of the five projects have a separate knowledge processing component that allows to define and maintain knowledge at runtime.

In the discussion it turned out that the remaining two research topics ‘Middleware’ and ‘Modeling and Exchange Specifications’ like IFC and XML are not actively explored, but are available technologies to be used by the researchers.

2.1 Main research objectives

Weimar project (Beer et al. 2004): The project participates in the priority programme since the year 2000. The goal of this project is the development of a distributed CAD system for the planning process.

The shared planning material of the project consists of a structured set of object versions (not documents) that is accessible over a network. In the real planning process, already stored object versions can not be changed for consistency reasons: Here, a planner can flexibly select subsets of these unmodifiable object versions and load them as newly derived object versions in his workspace for processing. It is generally known, that the development of a technical solution is an iterative process that often requires a long time: This long transaction ends by (1) discarding the results or by (2) storing the results as new object versions in the project.

Aachen project (Kraft and Wilhelms 2004): This project is part of the Priority Programme since the year 2003 and deals with the following problem:

Existing CAD systems mainly focus on the constructive design of a building. This approach very often results in an inconsistent product model instance because the CAD system does not know the semantics of the applied construction operations. The goal of this project is to support the conceptual design of a building in the early phase of product design. The conceptual design deals with semantic units (for instance rooms) and their relations: The conceptual model is consistent if the conditions defined by e.g. existing standards have been kept.

Instead of hard-coding the required knowledge by a programmer it can be defined at runtime by a knowledge engineer. The architect uses this knowledge to check the consistency of his conceptual design.

Wuppertal project (Huhn 2004): The project is funded since the year 2000. The main research objectives are:

The concepts of net-distributed planning processes in the field of complete building solutions with steel, wood, glass and metal (German: Komplettbau) are researched. For this application domain a software architecture for a distributed planning system is developed. This development is based on detailed use cases and a thoroughly investigated system design created by the UML language.

Net-distributed planning is realised predominantly by integrating already existing software systems: To fulfil the requirements of the new framework these systems have to be encapsulated by newly developed adapters (wrappers).

Dresden project (Weise et al. 2004): The project is part of the SPP 1103 since the year 2000. The project focuses on the distributed co-operation of a shared building instance of a standard product model.

The IFC has been chosen as the specific product model of the building instance. The additional description of the product model in the EXPRESS language makes the solution approach independent of the presently used IFC scheme or a special version thereof. Subsets of the shared product model can be checked-out and mapped on the product model used by the respective planner. Analogous to the Weimar project processing takes place inside a long transaction: Before check-in, the planner's product model instance is matched against the shared building instance: The changes as well as the versioning information are stored in the shared building instance. A future project goal is the merging of different variants of building instances.

Essen project (Schnellenbach-Held et al. 2004): The project participates since the year 2002. The goal of this project is the consistent handling of modifications of the product model in the planning process.

Analysing the consistency of a product model requires knowledge about technical standards and engineering know-how. Analogous to the Aachen project the knowledge can be defined at runtime by the planners in distributed knowledge bases. Once defined, this knowledge can be used for the consistency analysis of different product model instances.

2.2 Compliance of Basic Assumptions

Introduction: Although there exist many differences the solution concepts of the five projects have a great deal in common. All five projects use the object-oriented approach to create solutions for the distributed planning process in architecture and civil engineering. As network environment the Internet is accepted in its current form, where the limited bandwidth and the relative unreliability must be considered when designing a solution approach.

Product model: None of the projects is dependent on a special product model like IFC or STEP. The Dresden project can map arbitrary product models on EXPRESS. The Weimar, Aachen and Essen projects have their own proprietary product models and the Wuppertal project predominantly uses the product models of the integrated systems. The Aachen project has an additional implementation based on an available CAD model.

Conflict management: Distributed co-operation obviously requires a solution for conflict management. All projects agree in the assumption, that the conflict management provided by the ACID transaction components of available database systems (Date 2000) is inappropriate for the support of technical solutions in the planning process. The reason for this assumption is that during the iterative development of a technical solution user interaction can not be held outside the transactions: With long transactions the probability of conflicts is very high: With the ACID model conflicts were solved by long waits or by the abortion of transactions, the latter resulting in a loss of data.

System architecture: A unified system architecture of all five projects is shown in Figure 2. The architecture consists of two separate parts: The 'Product Modeling' part shown on the left side and the 'Knowledge Processing' part shown on the right side. Each part exists as a separate application. It should be noted that most traditional planning systems consist only of the 'Product Modeling' part shown on the left side and that the separate 'Knowledge Processing' part is very rarely seen.

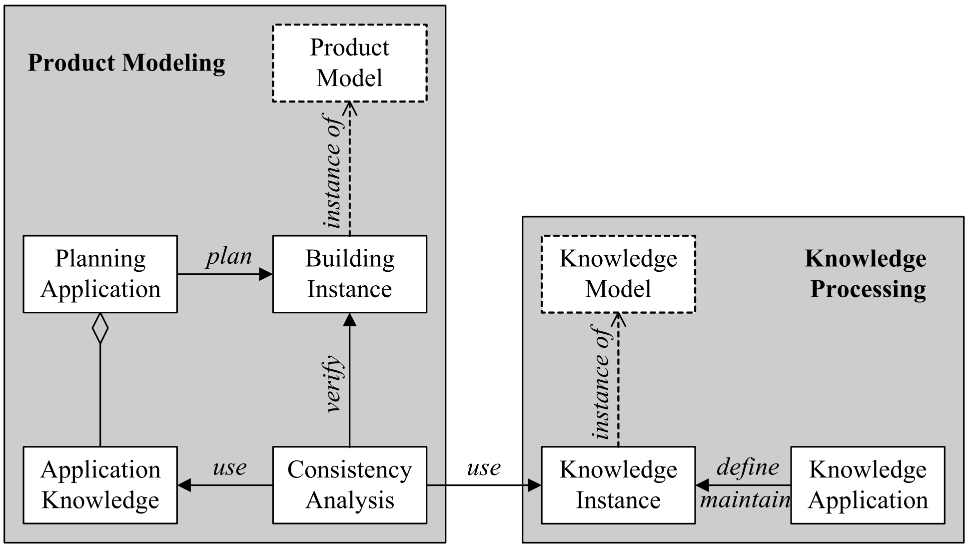

Figure 2: Unified System Architecture

The 'Planning Application' shown on the left side of Figure 2 is a tool for the architect and the civil engineer: The goal is to plan a building as instance of an appropriate 'Product Model' like IFC or STEP. The 'Planning Application' contains knowledge that has been defined by a programmer before compile-time. This hard-coded knowledge is denoted as 'Application Knowledge'. The 'Consistency Analysis' uses the 'Application Knowledge' to verify the consistency of the 'Building Instance'. This part of the system architecture is more or less found in all five projects of the workgroup.

The 'Knowledge Application' shown on the right side of Figure 2 is a tool to define and maintain knowledge at runtime. This part of the architecture is found in the Aachen and Essen projects: In the Essen project knowledge is defined and maintained by the planner whereas in the Aachen project a knowledge engineer is responsible for this task. The 'Knowledge Instance' is a specific instance of the 'Knowledge Model': Both knowledge models are proprietary developments – the specific concepts are explained below.

2.3 Product Modeling and Knowledge Processing

Introduction: The result of the planning process is a building instance of a certain product model. The consistency of this building instance is a common goal of the workgroup. However, different solution approaches are chosen to reach this goal.

Some projects have their knowledge defined in a separate knowledge base while other projects have it defined inside the application program. These differences will be discussed in more detail. The chosen product and knowledge models obviously have some differences. In order to compare the models against each other the specific model layering according to (OMG 2004) is considered. The model layers are shown in Figure 3.

Model layering: Two different modeling techniques are applied in the workgroup. Figure 4 shows the abstractions for the object-oriented modeling method on the left side: Layer M0 contains the objects that are instances of classes described on layer M1. The classes in turn are described on layer M2 by the UML language that is described itself on layer M3 by the Meta Object Facility (MOF) of the Object Management Group (OMG) (OMG 2004). Further abstraction layers do not exist – the MOF describes itself recursively.

	Metamodel Architecture
M3	Meta-Metamodel
M2	Metamodel
M1	Model
M0	Instance

Figure 3: Model layering according to (OMG 2004)

	OO-Modeling	Graph based Modeling
M3	MOF	PROGRES
M2	UML	Graph Schema
M1	Class	Node and edge types
M0	Object	Node and edge instances

Figure 4: Model layering applied in the workgroup

Besides the object-oriented model the Aachen project uses a second modeling approach, the so-called graph based modeling as a proprietary development. In the Priority Programme this model is applied for both the product model and the knowledge model: Layer M0 contains the node and edge instances of the graph for a specific application and layer M1 conatins the classes for these nodes and edges. The classes in turn are described by the 'Graph Schema' on layer M2. The task of the 'PROGRES' system on layer M3 is the specification and generation of the graph schema on layer M2.

2.3.1 Product Models

Weimar project: Object versions and their dependencies are stored in an instance of a shared proprietary product model. The planners can load arbitrary subsets in their workspaces. These workspaces are unversioned to (1) keep the complexity low and to (2) allow available CAD systems to be integrated and reused as front ends. The modeling approach is object-oriented with typical instances like shape and text objects on layer M0.

Aachen project: The conceptual design model of a building describes semantic objects and their relationships. A proprietary graph-based product model has been developed with typical instances like rooms and their relationships. The additionally developed ArchiCAD application naturally has been designed and implemented inside the object-oriented programming environment of the CAD system: Here, typical instances of layer M0 are room and relationship objects.

Wuppertal project: A virtual distributed product model is built by integrating the product models of existing planning systems: This is done by wrapping these systems and thereby implementing the interface required by the newly designed framework. Additionally, versioning information is stored by the framework. Typical instances on layer M0 are steelbeam and screw objects.

Dresden project: The shared product model is presently stored in IFC format. The additionally formulated EXPRESS-description allows the mapping of this IFC product model on other product models. This allows (1) the replacement of the IFC product model and (2) the mapping of checked-out subsets on the product model of the respective planner. Layer M0 consists of IFC-objects.

Essen project: A proprietary object model has been exemplary designed and implemented for the disciplines structural engineering and HVAC. The model allows the structuring of these objects according to application needs, for instance for design or proof. The instances of layer M0 are objects like walls and air ducts.

2.3.2 Knowledge representation

Weimar, Wuppertal and Dresden projects: These projects deal with knowledge that the programmer formulates by an object-oriented programming language. This kind of knowledge must be completely specified before compile-time and cannot be manipulated thereafter.

Aachen project: It is the task of the knowledge engineer to define and to maintain rules for the conceptual design of a building. Typical examples are attribute rules for the size of rooms or rules defining the relations and cardinalities between room types. The knowledge instance is defined and maintained by a graph-based tool. For instance, layer M0 contains rules for the attributes of or relationships between specific rooms.

Essen project: It is the task of the planning engineer to define and maintain knowledge. The knowledge base contains standards (for instance proofs and dimensioning) and planning know-how (for instance design knowledge). The knowledge is defined by object-oriented modeling as instances of classes like proofs, tables, diagrams and formulas that are well known by the planning engineers. For instance, a proof object on layer M0 can refer to other table, diagram or formula objects.

2.3.3 Consistency

Weimar project: A complete set of operations for the distributed co-operation of the product model has been defined: For each of these operations the consistency conditions have been formulated. The distributed system ensures that these conditions are fulfilled.

Aachen project: The conceptual design may be verified against the rules defined in the knowledge base at any time. If the design contains errors then the architect is informed about the rule that was not fulfilled and about the location where it appeared. In the future it is planned to continuously verify the consistency of the conceptual design.

Wuppertal project: Consistency is achieved by an accurate object-oriented design based upon reliable design patterns. Here, the approach is to create quality software by state-of-the-art design tools.

Dresden project: After check-out the iterative development of a solution by the planning engineer prohibits an overall consistency. However, consistency is achieved at check-in time: At these coordination points the shared product model instance and the planner's product model instance are matched and the respective changes are stored.

Essen project: After a modification of the building instance the necessary proofs are delivered by inference: The instances of the knowledge base can be dependent on each other and must be recursively evaluated based upon building and knowledge instances.

2.4 Co-operative Work and Conflict Management

Introduction: All projects support a net-distributed co-operation of the planning team. However, there are certain differences in the kind of co-operation. These differences can best be understood if co-operation is classified according to time (Bretschneider 1998) (Figure 5). Synchronous co-operation occurs at the same time while asynchronous co-operation takes place at different points in time. Three kinds of co-operation can be distinguished:

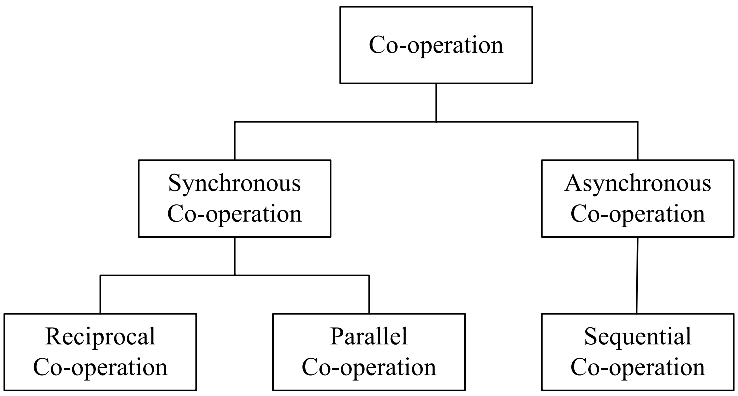

Figure 5: Classification of co-operation according to time

- **Reciprocal co-operation:** The members of the planning team work synchronously on the same task to achieve a common result. The task refers to a subset of the shared planning material that is manipulated synchronously by all members of the team. This kind of co-operation is predominantly addressed in Computer Supported Co-operative Work (CSCW).
- **Parallel co-operation:** The members of the planning team work synchronously on different tasks. Each task refers to another subset of the shared planning material. The objects of these subsets are independent from each other.

- **Sequential co-operation:** The members of the planning team work synchronously on different tasks. The tasks can refer to any subsets of the shared planning material. Unlike parallel co-operation these tasks are performed one after another in sequential order.

Weimar and Dresden projects: The building instance is manipulated in synchronous parallel co-operation. A pessimistic conflict management is implemented: Already stored objects of the building instance are not allowed to be manipulated directly. Instead, new object versions are derived and loaded into the private workspace of the planner where they can be manipulated (long transaction). Branches in the versioning graph are allowed: It is optimistically assumed that these variants can be merged later.

Aachen project: The knowledge bases are locally installed and can be manipulated by the knowledge engineer in single user mode. The distributed knowledge bases can be combined to a single knowledge base to be published on a central knowledge server. Many architects can simultaneously access the information from the knowledge server on a read-only basis. However, for the conceptual design of a building single user mode is appropriate: Formally, the architects work in synchronous parallel co-operation.

Wuppertal project: The Co-operation specification has been carefully designed by practical use cases: It turned out that a solution concept applicable in the area of complete building solutions would only be accepted if it supported a synchronous reciprocal co-operation. A framework has been designed to allow the reciprocal co-operation by integrating specific planning systems. Already existing planning systems must be wrapped before they can be integrated in the framework.

Essen project: Co-operation concerning the knowledge bases is very similar to the Aachen project: The distributed knowledge bases are replicated to a server where they can be simultaneously accessed by the planners on a read-only basis. The planning tasks are performed in synchronous parallel co-operation if independent subsets of the shared building instance can be formulated. Otherwise, the tasks are performed in asynchronous sequential co-operation.

3 Scientific Summary

Introduction: In this paper the research work of the five projects of the workgroup 'Distributed Product Models' has been described and discussed: Matches as well as differences of the solution approaches become apparent. The most important research results are summarized below.

Knowledge representation: All five projects deal with knowledge: However, there are differences concerning the definition and maintenance of this knowledge. While three projects deal with hard-coded knowledge the other two projects provide for knowledge processing at runtime. The latter approach allows the planning engineer, the architect or the knowledge engineer to define and maintain knowledge at runtime. This approach is more complicated but clearly has advantages over knowledge definition by the programmer before compile time.

Product models: Certain scientific results can be identified already. Although all projects deal with distributed product models none of them has committed itself to a specific product model like IFC: Important reasons for the missing acceptance are the complexity as well as the incompleteness of available product models. Specific implementations differ in quality and quite often new versions are released, thus making it necessary to replace the existing software tools and environments.

Three of the five participating projects use a versioned product model. Versioned product models allow the representation of revisions and variants of planning objects – a procedure that has proven in the planning process: Already stored objects are unmodifiable and thus remain consistent. Deriving new object versions for modification makes the evolution of the planning process traceable and allows a synchronous parallel co-operation of the shared planning material even if interdependent objects are processed.

A building instance is a structured set of objects or object versions. In a distributed environment the planners select subsets of the building instance that are loaded into a workspace for processing. The flexible description of consistent subsets has been identified as a central research objective in three of the five projects.

Another central research topic is the consistency of building instances that becomes possible in a shared distributed environment: Different methods like versioned product models, explicit representation of object dependencies, description of conditions to be fulfilled by operations, and well designed quality software have been proposed as solution approaches so far.

Acceptance of existing technologies: The Internet in its actual form (relatively slow and unreliable) is accepted as network environment by all five projects of the workgroup. Available (single user) planning systems are reused in a distributed environment, for instance by wrapping patterns (Wuppertal) or by providing new operations for the users (Weimar).

Conclusions: The field of research in distributed product modeling is a very wide one. In the projects involved different objectives are in the focus of research, like versioning and knowledge processing. These research objectives do not exclude each other but will be the basis for a consistent distributed process modeling in the future.

4 Acknowledgement

The author gratefully acknowledges the support of this contribution by the members of the workgroup ‘Distributed product models’ within the scope of the priority programm ‘Network-based Co-operative Planning Processes in Structural Engineering’ (SPP 1103) funded by the Deutsche Forschungsgemeinschaft (German Research Foundation).

5 References

Beer, D. G., B. Firmenich, T. Richter, and K. Beucke (2004). *A Concept for CAD Systems with Persistent Versioned Data Models.* In digital proceedings of ‘Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)’. Weimar: Bauhaus-Universität Weimar.

Bretschneider, D. (1998). *Modellierung rechnerunterstützter, kooperativer Arbeit in der Tragwerksplanung.* Düsseldorf: VDI Verlag.

Date, C. J. (2000). *An Introduction to Database systems. Seventh edition.* Reading: Addison-Wesley.

DFG. *Deutsche Forschungsgemeinschaft (German Research Foundation).* <http://www.dfg.de> (13 April 2004)

Huhn, M. (2004). *Abstract and Concrete Scenarios in Concurrent Engineering.* In digital proceedings of ‘Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)’. Weimar: Bauhaus-Universität Weimar.

Kraft, B., and N. Wilhelms (2004). *Interactive Distributed Knowledge Support for Conceptual Building Design.* In digital proceedings of ‘Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)’. Weimar: Bauhaus-Universität Weimar.

Meißner, U. F., and U. Rüppel (2004). *DFG Priority Program 1103: Network-based Co-operative Planning Processes In Structural Engineering.* <http://www.iib.bauing.tu-darmstadt.de/dfg-spp1103/en/index.html> (13 April 2004)

OMG. *Object Management Group.* <http://www.omg.org> (13 April 2004)

Schnellenbach-Held, M., M. Hartmann, and T. Pullmann (2004). *Knowledge Based Systems in Distributed Design Environments.* In digital proceedings of ‘Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)’. Weimar: Bauhaus-Universität Weimar.

Weise, M., P. Katranuschkov, and R. J. Scherer (2004). *Generic Services for the Support of Evolving Building Model Data.* In digital proceedings of ‘Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)’. Weimar: Bauhaus-Universität Weimar.