Information Management in the Concurrent Design Process*

Rainer Wasserfuhr[1] and Raimar J. Scherer[2]
Institute of Structural Mechanics and Computer Science in Civil Engineering, TU Dresden, Germany

1 Introduction

Former efforts in the integration of design information management have concentrated on interoperability of applications like e.g. CAD, structural analysis or facility management, applying product data technology with product models introducing additional application independent model layers.
One of the most advanced efforts in this directions is the STEP methodology, providing terminology, formal modeling languages, database and physical file exchange interfaces and existing standards for different application areas (Application Protocols).
In the last years it has become clear, that besides interoperability of autonomous applications, the concurrent processes of model instantiation and evolution have to be modeled, including the relationship to available project resources, persons, legal requirements and communication infrastructure. These needs are also driven by increasing importance of client server architectures, which are already used in other engineering domains, but now are infiltrating the SME dominated A/E/C sector, because former investment risks of introducing new IT environments have decreased.
This paper discusses some basic concepts for an emerging methodology relating the fields product modeling, project management and workflow systems by elaborating the concept of a *process model*, which gives a decomposition of the project goals into executable activities. Integrated, client server based information management systems should be related to process models to detect pending activities, deadlocks and alternatives of execution.
The paper is organized as follows: In chapter 2 we give a short overview on current information layers in product modeling, then introduce the basic concepts of process models and show, how product models can be interpreted from a process model point of view (Chap. 3). An abstract system architecture is sketched (Chap. 4), including an implementation scenario supporting Internet services (Chap. 5).

2 Product Modeling Methodology

In [FF96] construction information of product model based environments is classified into the layers of data modeling language, conceptual layer, information layer and the layer of real objects, as shown in Table 1.

The information generated during one project is related to predefined data and concept definitions, which are themselves represented as a relational system of concepts, connected via specialization, part- of- relationships (partonomies) and aggregations.
An example for logical references across the information layers is given in Figure 1.

Layer	Examples	Formally represented as	Size (appr.)
Data Modeling Layer	EXPRESS	Extended Backus-Naur Form (EBNF) of ISO 10303-20	5 pages
Conceptual Layer	Building Construction Core Model (BCCM) / Industry Foundation Classes (IFC)	EXPRESS	40 pages
Information Layer	Information of a concrete building project, e.g. CAD+document data of the New Munich Fair	STEP Physical Files/ Commercial CAD data	30 GigaByte
Reality	New Munich Fair	N/A	N/A

Table 1: Classification of Information Layers

(1) Dipl.-Inform. (2) Prof. Dr.-Ing.
E-Mail: `{Rainer.Wasserfuhr|Scherer}@cib.bau.tu-dresden.de`
`http://wwwcib.bau.tu-dresden.de`
(*) This research study was founded by *Sächsisches Ministerium für Wissenschaft und Kunst* under grant no 4-7541.82-0370/574.

Product models with a high degree of matureness have already been developed in several research projects, like ATLAS, CIMsteel, COMBI or COMBINE. They resulted in standardization efforts like the Building Construction Core Model (BCCM) or STEP AP 225. Consequently, also in the A/E/C sector, product data technology (PDT) is currently on the way to commercial products, of which the Industry Foundation Classes (IFC) are the most prominent ones. We assume them to be represented in a formal STEP/EXPRESS representation and we will enrich these models with the necessary design process related semantics.

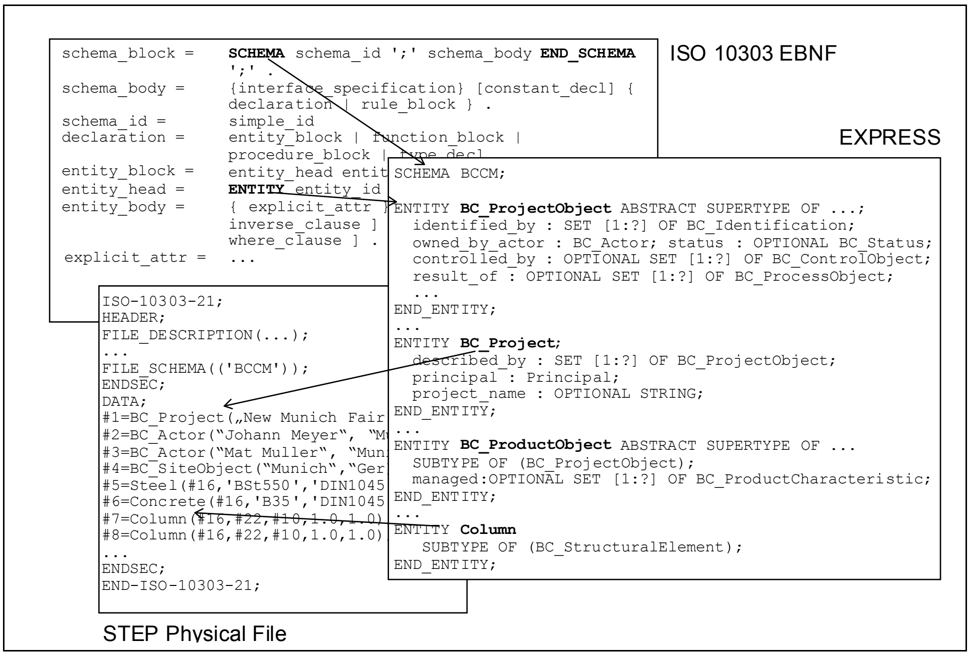

Figure 1: Formal Representations for the different Information Layers

Further discussion and open problems of the interoperability across the different layers resp. between applications can be found in [KS96], but will not be elaborated further here.

3 Concurrent Design Processes

Computer integrated design processes can be described on a scale ranging from a macro to a micro layer, as depicted in Figure 2.

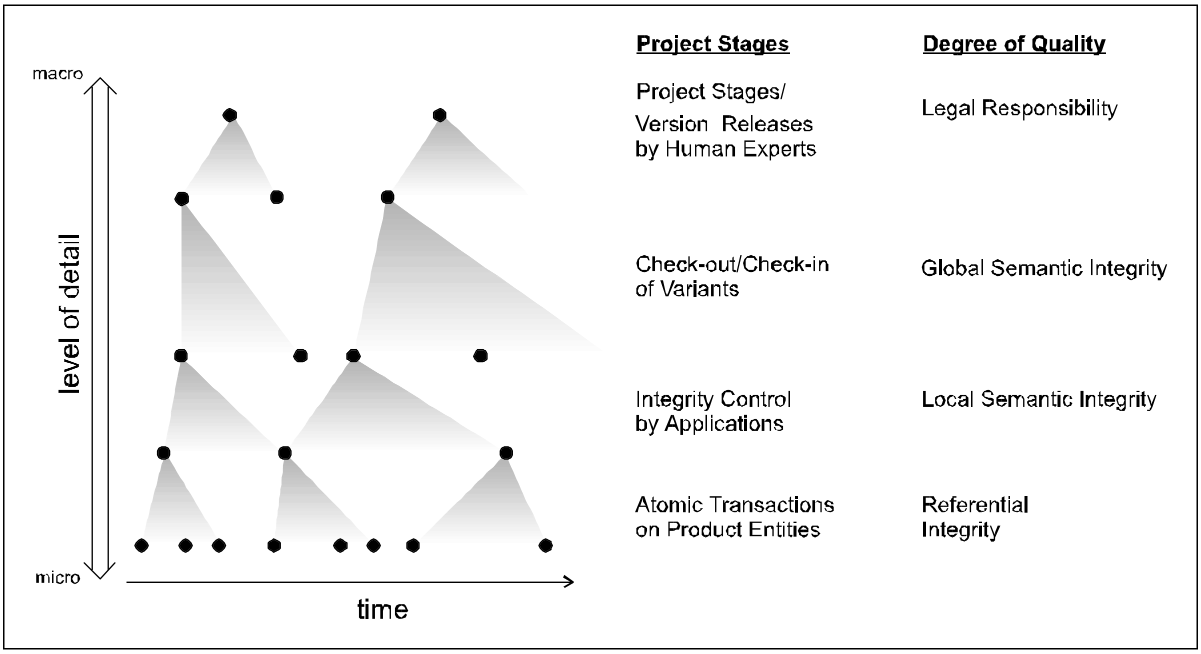

Figure 2: Granularity of Process Activities

The overall design process is understood as a successive transformation of design requirements into a final design solution, consisting of design decisions, which determine the form of the final product. Additionally, a project also includes the acquisition of the requirements themselves. This can be modeled as requirement refinement, but also sometimes as a requirement revision, forcing a backtracking from initial requirement

assumptions. Due to the inherent complexity of the construction area, the design process is splitted into domains of different design aspects, which are developed in parallel. The final design solution has to fulfill the requirements of all of them. The design process is a minimization of the inconsistencies which appear in the different *views,* such as customer needs, structural laws, design for construction, costs and aesthetics.
The design process can be characterized as a combination of both human and computational problem solving on different levels of complexity, where e.g. design rules and experience of human experts are used, but also results of numerical and geometrical reasoning or cost computations.
Execution of design activities is based on input and output of design information. Design activities are executed by implicit or explicit reference to a limited set of data which is related to the current state of the design solution. Current practice for the encapsulation of input and output of design data are documents and plans. Integrated process and product model based systems should be able to refine this to a better level of granularity.

We summarize this understanding of design processes by applying a prominent building block for generic process descriptions. Therefore we use the ICOM boxes of the IDEF0 methodology, consisting of:

- **Inputs:** Resources necessary to carry out or complete the given activity
- **Controls:** Constraints on the activity like the consistency in the views of different experts
- **Outputs:** Resources produced or modified by the activity
- **Mechanisms:** Tools (users, applications, equipment, etc.) required to accomplish the activity

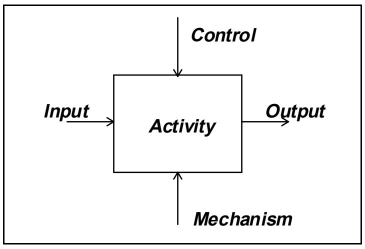

Figure 3: IDEF0 Box

High level product IDEF0 activities ant their interrelations can be obtained from the Application Activity Model (AAM) of the STEP approach. This model describes, to which resp. from which other applications the information of an application is exchanged. According to our client-server approach, we assume, that the design information of the applications is continuously managed by a central product data management system (PDMS), which covers any project information flow and generates the application specific views from the central product model, used and updated by the applications. Additionally, it is responsible for concurrency and access control for the shared project database and the workflow management.
The design processes are electronically supported by the PDMS (Figure 4). One basic task of the PDMS is the maintenance of the valid electronic information model versions, which are released by human experts. This legal validity of a version might be indicated either by a signature of a printout paper of a plan or document or possibly with electronic signature directly on the information level.

The valid versions of design information model of a project can be generated by logical views from the product model. It should be emphasized, that the PDMS not only maintains the pure data which is necessary to generate the printed documents, but also the semantical structures behind the resulting document.

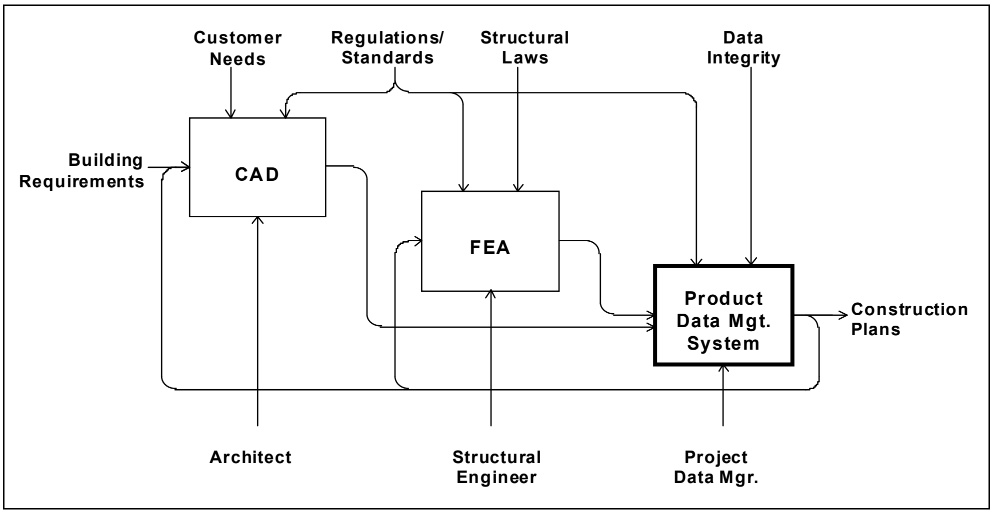

Figure 4: Top level information exchange activities

Concurrency Limitations caused by Transaction Principles: The *atomic operations* on product data are transactions modifying or querying single entity attributes. The IDEF control constraint on this level is basically referential integrity. Currently, most of these operations are local to one application model. Deeper semantic control is obtained on application level. E.g. collision-freeness of 3D geometric data is currently usually checked on the level of local applications.

These operations may be executed concurrently, as long as the transaction principles of database theory [Wei88] are not violated. In current practice, there is no explicit semantic integrity control across applications. A higher level of semantic quality can be obtained by unification of partial models, see e.g. [KS96] for a knowledge based approach. The degree of semantic integrity also depends on the depth of which design rules are modeled and checked. E.g. [VBHJ+96] proposes a hybrid representation style including design rules, exemplified for designing buildings with complex installations.

Workflow Management: Workflow Management become a popular concept in many domains. The basic task of workflow systems is the electronic acquisition and management of *worklists* for the different human actors [WFMC94]. The traditional workflow approach has grown from domains with many atomic but similar activities, like e.g. bank accounting. It implies a relatively homogeneous form of activity execution. For concurrent design environments, we have to assume a much more complicated data model, requiring deeper knowledge about the cooperatively developed product. In our context, we therefore choose an extension of the transactional product data management into the following two dimensions:

1. Worklists can be generated in a straight forward manner from the concurrency control of the project database. E.g. if user A wants to access data which is currently locked by user B, the system adds a pending activity to the worklist of A and a request for unlocking on B's worklist. If B unlocks the data, the pending activity on A's worklist is reactivated. Circular locks, so called *deadlocks*, can be detected by the system.
2. The system also enables users to indicate their intentions to access or request product data which does not yet exist. The system automatically generates an empty "information container" which is preserved until the requested actors will fill it. Again, indication of intentions covers the whole time scale, starting from atomic transactions, only lasting minutes, up to the expected deliverables of whole project stages.

Micro level workflows are complemented by predefined high-level process definitions. By default, the worklist system is set up by common design process patterns.

Quality of Design Information: Besides the commonly used definition of concurrency as the simultaneous execution of activities, we also focus on a *quality oriented understanding* of design activities. In this perspective, design activities should be executed with maximal forecasting of the consequences w.r.t. the life-cycle aspects of the product, as controlled by the different model views, and also consistently operate across the different levels of information model transformations.

4 System Architecture for Integrated Information Management

Concurrent Object System: The product data management system currently under development is implemented as a system of objects which are communicating concurrently via method calls. Object communication can be triggered by user requests or by periodic update computations started by the system itself. Objects can be declared to be persistent, which implies, that they will survive server shutdowns. By using a virtual memory architecture, it is achieved, that the amount of persistent object data may increase the actual main memory. The required objects are loaded dynamically from secondary memory, if they are needed.

Objects may reference each other. Any persistent object of the system is reachable from a dedicated persistent *root object*. The object system constitutes a connected *reference graph,* also shown in Figure 5. Reachability of any object is one of the basic invariants of the architecture which has to be preserved before and after any operation on objects. Any persistent object preserves referential and state integrity before and after a method call. Thus, the system architecture fulfills the basic reliability requirements of a shared database system.

The object system provides authentication of users, logging of activities, access control and condition based notification of changes.

Concurrency Enabled Applications: The traditional approach to the integrate applications into product model environments is based on common data models, where the application data definitions are first captured in the Application Reference Model (ARM), which is then embedded into the STEP generic resources, resulting in the Application Interpreted Model (AIM). The applications have to provide translators for the conversion of their private data to and from STEP physical file format resp. an SDAI interface.
In a concurrent engineering setting, additionally a behavioral description is necessary, which specifies the method calls, that the applications can execute on the concurrent object system.
A formal framework for extending EXPRESS with behavioral definitions is given in EXPRESS-P, see [Mül95] and will also be integrated in version 2 of EXPRESS. Another mature approach is the CORBA Interface Definition Language (IDL, see [OMG95]), which is a generic framework for the definition of distributed applications, but this approach is currently not directly linked to the STEP methodology.

5 Implementation for Internet Environments

Finally, a brief outline of the system architecture shows, how the information management system can be accessed using conventional Internet technologies. Internet technologies are especially relevant for inter-organisational cooperation. Internet technologies and standards like HTML, HTTP, CGI, SMTP (see [BLCG92]) or Java have also brought new dynamics into the development of intra-organisational networks, bridging the problems of heterogeneous platforms, operating systems or networking technologies. The Internet gateway implemented here is especially designed to manage project member data, their worklists, document databases, and product model browsing.
The World Wide Web (WWW) family of technologies is a multi-client multi-server architecture. Current shortcomings of this architecture for concurrent engineering architectures result from the initial WWW design goals which were mostly restricted to stateless data.

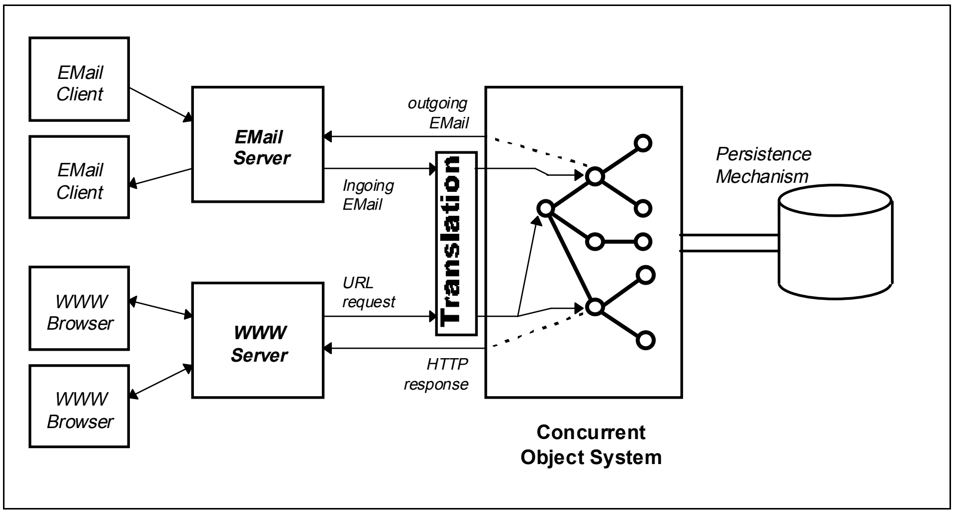

Figure 5: Integration of Internet Services

In contrast to standard HTTP servers, any request is therefore computed at request time. *Sessions* identify successive requests of one user and allow a server side history management and the configuration of user preferences for a session.
HTTP requests are translated into the respective method calls of the concurrent object system. This can be achieved either by a

- server side reimplementation of the HTTP protocol or by
- a dedicated CGI script of an existing HTTP server, which uniformly decodes the URL parameters, forwards the request to the object system and awaits the object system response.

HTTP responses contain HTML references with encoded object IDs and method calls. Whole documents can be transferred both to clients and to the server. Transfer to the server is based on the “file upload“ mechanism. HTTP responses may also contain non-HTML document types like VRML or commercial CAD file formats. The file types are represented with the standardized MIME file type mechanism.
Client side helper applications may be used to view and manipulate those file types directly by the user.
The EMail server is implemented similarly. Ingoing Emails may contain document attachments or method calls similar to commands of mailing list servers.

6 Conclusion

Information systems in the A/E/C sector are rapidly growing together from stand-alone applications to client-server and client-network environments. While product models first have been introduced to keep as much semantics as possible during data exchange, the exchange activities and operations on product entities themselves are now getting into to focus of quality oriented process models.

It is suggested to describe process activities both bottom-up from atomic transactions on product entities, applying the transaction concepts of database theory, and top-down from overall project stages and releases of document versions. Workflows are dynamically acquired, according to the information demands of the design actors. The implementation of a PDMS can be achieved by extending the object oriented paradigm with a concurrent communication mechanism. Synchronisation mechanisms are then derived from the requirements of state integrity and reachability of objects in the reference graph. Internet services may be integrated into the object system by a generic request translation mechanism.

7 Acknowledgments

We gratefully acknowledge the financial support for this work by *Sächsisches Ministerium für Wissenschaft und Kunst* under grant no. 4-7541.82-0370/574. We would also like to mention the fruitful discussions with Peter Katranuschkov, which have influenced this paper.

8 References

[AMS92] R. Anderl, B. Malle, and M. Schmidt. *Concurrent engineering based on a product data model.* In Proc. CALS EUROPE '92, pages 145-154, Paris, May 1992. Hermes.

[BLCG92] T.J. Berners-Lee, R. Caillau, and J.F. Groff. The world-wide web. *Computer Networks and ISDN Systems*, 25:454-459, 1992.

[FF96] M. Fischer, T. Froese: Examples and Characteristics of Shared Project Models. *Journal of Computing in Civil Engineering*, ASCE, July 1996, 174-182.

[ISO92] ISO 10303-11. The EXPRESS language reference manual, ISO TC 184/SC4, Gaithersburg, Md. 1992.

[Jab95] S. Jablonski. Workflow-Management-Systeme: Motivation, Modellierung, Architektur. *Informatik Spektrum* 18(1): 13-24, 1995.

[KS96] P. Katranuschkov, R. Scherer. *Schema mapping and object matching: A STEP-based approach to engineering data management environments.* In CIB-W78 Workshop: Construction on the Information Highway, pages 335-344, Bled, 1996.

[Mül95] W. Müller. *The Process Modelling Language EXPRESS-P.* ISO TC 184/SC4/WG5 N230 document 1995, ftp://ftp.uni-paderborn.de/unix/express/express-p/n230.ps.Z, 127kBytes.

[OMG95] Object Management Group: *The Common Object Request Broker: Architecture and Specification, Revision 2.0,* July 1995, http://www.omg.org/corba2/cover.htm.

[VBHJ+96] A. Voß, B. Bartsch-Spörl, L. Hovestadt, K. Jantke, U. Petersohn, G. Strube, FABEL, *Künstliche Intelligenz* 10(3):70-76, 1996.

[Wei88] G. Weikum. *Transaktionen in Datenbanksystemen.* Bonn : Addison-Wesley, 1988.

[WFMC94] Workflow Management Coalition. *Terminology & Glossary- A workflow management coalition specification.* Technical report, June 96, http://www.aiai.ed.ac.uk/WfMC/DOCS/glossary /glossary.html, 62433 bytes.