CONCEPTUAL MODELLING OF A CONCURRENT ENGINEERING ENVIRONMENT

Z. Turk[1], R. Wasserfuhr[2], P. Katranuschkov[3], R. Amor[4], M. Hannus[5] and R.J.Scherer[6]

ABSTRACT

Since the late 1980s it has been suggested that product modelling is the key to computer integrated engineering, which is also a prerequisite for computer assisted concurrent engineering. Later experiences have shown that product modelling alone is not sufficient and that other aspects, such as processes, documents etc. have to be modelled as well. In this paper we suggest *environment modelling* as an approach which integrates all kinds of construction information, such as the information about products, processes and documents. We propose an environment modelling framework which decomposes an abstract concurrent engineering environment into several modelling spaces. For practical reasons we use a two dimensional orthogonal framework decomposition. Along the first axis the framework is decomposed according to the modelling aspect (construction, information system, generic), and along the second axis according to the level of detail (neutral, aspect, application). In this paper we present the decomposition criteria, the resulting framework and some key components of the environment. Reported is the work in progress performed as part of the EU ESPRIT project ToCEE.

Keywords – product modelling, process modelling, concurrent engineering, computer integrated construction

1. INTRODUCTION

In this section we describe the context of the research, outline related efforts, and state the problems focused upon in our research work.

1.1 Towards a Concurrent Engineering Environment

Winner et al. (1988) define concurrent engineering as "*a systematic approach to the integrated, concurrent design of products and their related processes, including manufacturing and support. This approach is intended to cause the developers, from the outset, to consider all*

[1] Faculty of Civil and Geodetic Engineering, University of Ljubljana, Slovenia, ziga.turk@fagg.uni-lj.si

[2] Applied Computer Science in Civil Eng., Technical University Dresden, Germany, rainer@cib.bau.tu-dresden.de

[3] as 2, peter.katranuschkov@cib.bau.tu-dresden.de

[4] IT Research and Application, Building Research Establishment, UK, trebor@bre.co.uk

[5] VTT, Technical Research Centre of Finland, matti.hannus@vtt.fi

[6] as 2, scherer@cib.bau.tu-dresden.de

elements of the product life cycle from concept through disposal, including quality, cost, schedule and user requirements". Dean & Unal (1992) define concurrent engineering as "*getting the right people together at the right time to identify and resolve design problems. Concurrent engineering is designing for assembly, availability, cost, customer satisfaction, maintainability, manageability, manufacturability, operability, performance, quality, risk, safety, schedule, social acceptability, and all other attributes of the product*".

To an extent, concurrent engineering is the way the building industry operates. However, the quality achieved through the application of concurrent engineering in real design and construction practice differs remarkably from case to case. Existing engineering software tools provide little support for concurrent work. The achieved level of concurrency is lowered by differences in the potential for concurrent engineering work in the different enterprises. Organisational patterns required for intensive concurrency are being re-invented over and over, because buildings are one-of-a-kind products and their construction is managed typically according to the technical aspects of a project, and not according to the business processes in an enterprise.

ToCEE (Towards a Concurrent Engineering Environment in the Building and Engineering Structures Industry) is a three year ESPRIT project (1996-1998). ToCEE aims to contribute substantially to the improvement of the organisational infrastructure for concurrent engineering by utilising the opportunities offered by today's IT. The primary objectives of the ToCEE project are the development of an overall conceptual framework, as well as specific software tools, for concurrent engineering support. Some of these tools are intended to be applied directly in the current software implemented by end-users at the end of the project. Others will be developed and tested as prototypes in ToCEE leading to a more long term development.

ToCEE aims at the integration of the different models which represent a project, and the different modelling and information management aspects that are present in a building project, in order to increase substantially the power and the quality of concurrent engineering operability. The overall goal is to accumulate the whole spectrum of different development efforts into a conceptual framework and develop a prototype environment for concurrent engineering with very general applicability.

1.2 Related Work

In the application of IT for concurrent engineering in the building industry valuable steps have already been achieved. During the second half of the 1980s a number of research papers from Eastman(1988), Gielingh(1988), Turner(1988) and others introduced the "product modelling" approach into construction and suggested it as a key technology for construction information exchange and computer integrated construction.

Product model based integration is maturing into a set of international standards known as STEP (ISO 10303). Several EU projects like ATLAS (Boehms and Storer, 1994), CIMsteel (Watson and Crowley, 1995), COMBI (Scherer and Sparacello, 1996), and COMBINE (Augenbroe, 1995) have developed prototype product model environments which have proven the principal validity and effectiveness of the approach and have greatly stimulated further research, standardisation and implementation activities. Today product modelling has also become the basis for the IFC (Industry Foundation Classes) technology developed within the industrial IAI initiative.

Valuable results have been achieved in process modelling, e.g. by the CALS initiatives (Henerson, 1991). The IRMA model (Luiten et al, 1993) was one of the first attempts to join the product and process models into a unified framework.

Workflow developments and electronic document management has been investigated by the WfMC (Workflow Management Coalition) as well as in several commercial (IIC Consulting and Cimtech Ltd., 1996) and academic (Amor and Clift, 1996, Turk, 1995) electronic document management systems.

1.3 Problem Statement

As outlined in the previous section, there has been a growing awareness of the importance of engineering data management. Engineering databases, fitting into a schema defined in standardised conceptual product models, have been regarded as one of the key components of integrated construction. However, later research has acknowledged that the modelling of only products is insufficient.

Despite all these efforts and achievements what is still lacking is - according to our opinion - the treatment of all information handled by concurrent engineering systems (product information, process information, documents, people, software, hardware) not as single, mutually independent or only partially interrelated aspects, but as a whole consistent system for concurrency. Therefore this system's interoperability and co-ordination is of utmost importance.

In this context, the problems related to product modelling include:

a) Unless software exists, which would implement the model, it is difficult to objectively evaluate the usefulness and correctness of the designed model.

b) For computer integrated construction, product data is important, but is not the only data set required. It may also not be the "centre of the universe" in the sense that all integration should happen around something as complicated and difficult to agree upon as product models appear to be.

c) Information exchange and sharing symbolised as "data in once" is important, but is not the only requirement which integrated environments should solve. In addition, concurrent engineering environments must support parallel work, version handling, conflict management, document management, information logistics, building regulation management etc.

We propose that product, process, document, regulations and legal aspect (to name just the key components) modelling should be part of a much wider framework which we call "*system modelling*". System modelling is a step in the design of software components supporting concurrent engineering. The focus of the research and of the modelling are thus not the construction products and processes themselves, but software systems which may support the advanced production methodology understood as "concurrent engineering".

During the early stages of ToCEE we have identified the following specific problems related to system modelling:

a) The Modelling scope is even larger than with product/process modelling since both are included together with many additional aspects to be considered.

b) Knowledge about the domains is dispersed among several disciplines. In addition to several branches of construction, generic information-technology such as database, distributed objects, networking and communications concepts should be modelled.

c) Modelling is intimately related to the individuals' understanding of the modelling domain and is therefore difficult to agree upon.

The approach proposed in the next section addresses these problems.

2. MODELLING FRAMEWORK

In this section we present an environment modelling framework which brings together the process, product, document and regulation models and puts them into relation with the meta data, resulting in a coherent matrix which enables distributed design of concurrent environments. A suggested terminology, the decomposition criteria and the actual decomposition matrix are presented and discussed.

When we have been trying to bring several such concepts into one coherent framework we have found out that the product/process modelling approaches are too narrow, because they concentrate on construction domain aspects of the material world and neglect the information aspects of a computerised environment. An engineering environment, however, needs to know about both, and therefore both must be formally modelled.

2.1 Terminology and Domain Overview

Concurrent engineering (CE) is a systematic approach to the integrated, concurrent design of products and their *related* processes, including manufacturing and support. A *Concurrent engineering environment* (CEE) is where concurrent engineering takes place e.g. in a specific company, design office or in a virtual enterprise. This environment includes the company's staff, premises, knowledge and other resources and may also include any other items which are related to its activities (partners, clients, government etc.). *CEE system* is an IT supported system which enables CE.

Within a CEE, basically three types of information are generated and manipulated:

1) *The information about the constructed facilities*, including construction products, construction processes, construction regulations and codes, requirements etc. Information in this section is about material items only and is not associated or influenced by information technology. E.g. when we model a reinforcement bar we are not interested at all how this information is stored in documents, databases or how it is versioned.

2) *The information about the CEE system itself*, including information products (documents, databases) and information processes and the environment such as hardware, software, server, client, person, application, organisation etc. Concepts of this type are largely generic. They relate to construction concepts (e.g. document types), but the key features of the concepts are domain independent.

3) *Information about the information.* Concepts like information ownership, authorship, versioning, configuration and the even more abstract object model common to all components of the information system.

A CEE system stores information as *CEE data*. This data is stored in various types of databases such as relational database, object database or in a file system "database". The CEE data fits into a *CEE schema*. The CEE schema has several components, or sub-schemas, which fit into *a*

CEE modelling framework (CEEMF). Because all components of the schema are interrelated, we use the term “schema”, not plural “schemata”.

The process during which certain types of CEE are abstracted into a CEE schema is called *modelling*. A process during which a CEE system is built is called *software development*. A process during which the databases are populated with specific data is called *construction*.

CEEMF defines how to decompose the CEE information, i.e. it defines the decomposition principles, the decomposition itself, main features of each of the components and the general interface they have with other components.

2.2 Decomposition Criteria

Before actually looking into how to organise the CEE system information, we shall define certain information decomposition criteria.

- The decomposition should simplify the modelling and consequently the implementation of the CEE without sacrificing features of the CEE system.
- The details of each component should be largely autonomous and can be developed with little reference to other cells.
- The information within a component may have many references of various type (uses, used-by, specialises-into, part-of, ...) to information within the same component.
- The information within components should have few references to objects in other cells. All references should be, as much as possible, of the same type. For example a chunk of project information will always reference (use) product information onto which this chunk of project information relates to, but not vice versa.
- The decomposition should follow knowledge domains. Components should try to separate general knowledge from domain specific knowledge, such as structural engineering, so that experts in each field can easily contribute to developing each of them (e.g. it is wise to treat a construction product - in this framework - differently from a software product or a CEE system product - such as a document - but make sure they both *are* products).
- The components should use, rather than include knowledge which is not component specific (e.g. products need not know about the documents in which the information about them is presented and archived).

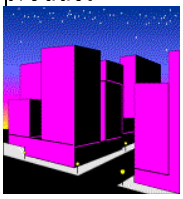
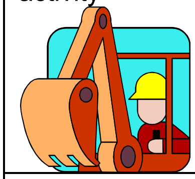
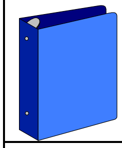
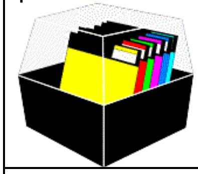
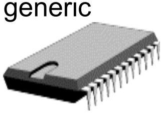
Deep “is-a” class hierarchies are undesirable, because, at the implementation level, they tend to bind more tightly than “uses” relations.

2.3 Decomposition Matrix

Table 1 shows the decomposition matrix with some sample objects. Basically the information is divided (row-wise) into construction specific information, CEE system information and generic concepts. Construction information is further decomposed into product, process, requirement and regulations information. Similarly, CEE system information is decomposed into CEE system product (which is information) and CEE system process (during which the information is created). Construction product and process information is further decomposed into reference, neutral, aspect and application models.

Figures 1 and 2 show a system’s process and information model. Bjoerk (1996) used the abstraction of either material processes or information processes. Material processes change the material world (e.g. pouring concrete into the formwork). Information processes process information (e.g. determine how much concrete do we need, when do we need it etc.).

Table 1: ToCEE Modelling Framework decomposition matrix.

information type	**information subtype**	**sample concepts**	**further decomposition**
construction information	product 	project, construction product, construction component, material, building, site, element, connector, space, wall, slab, equipment, pipe, duct, opening, door, window, ...	by specialisation (neutral, aspect, application)
	process / activity 	construction process, person, organisation, work section, work group, task ...	by specialisation and by building life cycle stages (feasibility, design, planning, construction, ...)
	document 	bill of quantities, drawing, contract ...	
	legal	regulation, requirement, clause, provision ...	
CEE software system information	product 	information chunk, file, message, generic document ...	
	process / activity	client, server, actor, user, application, communication event, request, reply, input, output, flow, job, ...	
	legal	digital signature, authentication, watermark ...	
generic 	meta information	owner, lock, version, change, configuration, view, authorisation, access right ...	
	object model	model, schema, class, object, attribute, method, relation ...	

Information processes (1) control the material processes and (2) monitor output of the material processes – thus in effect the material processes are supplying feedback information to the information processes. The CEE system is an information process which supports, with IT, both the information and material related construction processes.

Figure 2 details relations between key concepts in the CEE system. The key product of the information processes are “information chunks”. We define them as “smallest set of data with its own meta-information". The data in an information chunk is about construction products or construction processes which fit into schemata defined in the first four rows of table 1. The meta information is about who, when and how created this information and how is this information related to each other. Examples of information chunks can be as high level concepts as documents or simple entities like objects of even object attributes. Information chunks are input-to and output-of information processes which are performed by actors who may be people or applications.

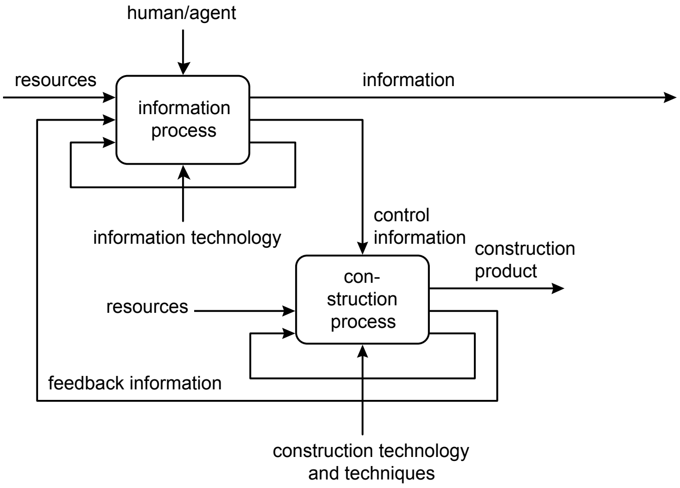


Fig. 1: CEE system's reference process model.

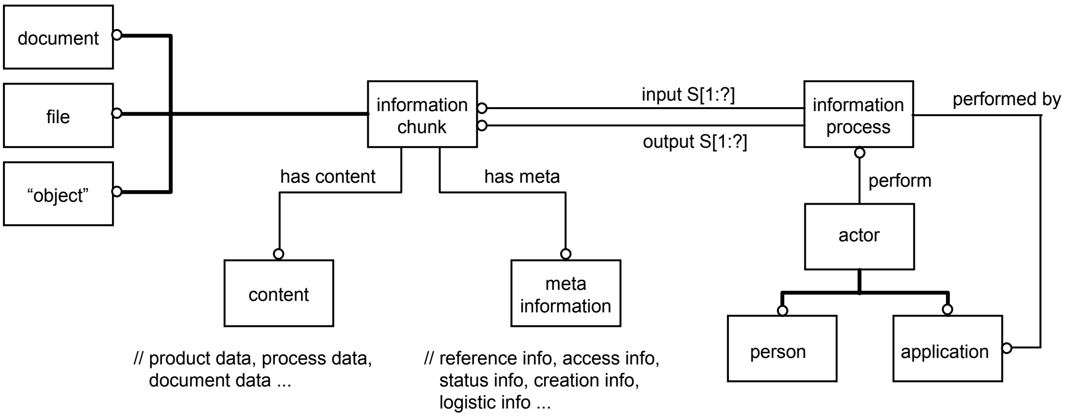


Fig. 2: CEE system's reference information model.

3. ARCHITECTURE

3.1 System Architecture

Different subsystems of the CEE system have to be combined, with specific emphasis on increasing the operability of concurrent engineering. On the system level, just like on the modelling level, we do not think that any of the subsystems (e.g. process management or product model) should have a central, special role. Instead a Ñliberal" distributed architecture will be applied together with an underlying meta model.

The system architecture follows the client server paradigm, starting with common agreements on interface definitions and then allowing a parallel system development for the client and server side. In this approach, each system is modelled as a server or set of servers. They are connected by a neutral information logistic system (ILS), which is modelled as a server as well. The main task of the ILS to handle information-chunks, moving them from server to server.

Any project communication is primarily performed by client requests to an information logistics server, who's role is to identify the correct target server and forward the request. Additionally, the information logistics server provides a centralised logging, access control and serialisation of concurrent requests. This type of system design enables the clients to operate in an integrated space of services, independent of the physical location of a server side service.

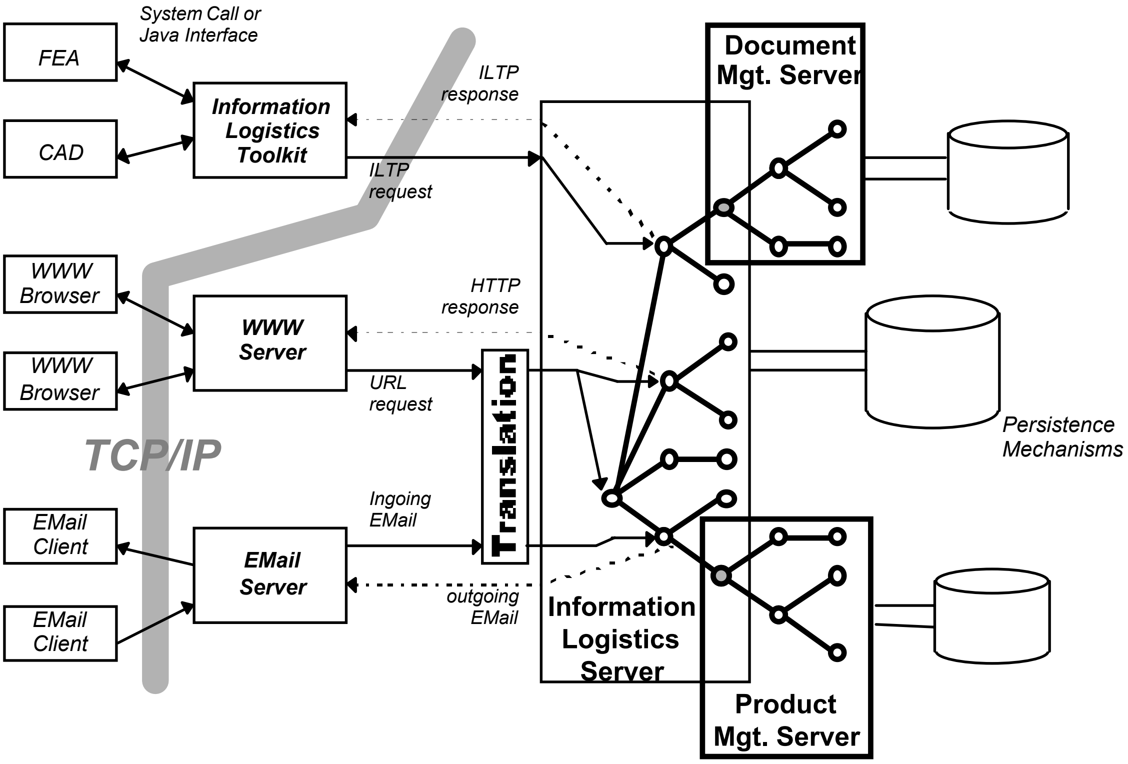


Fig. 3: System architecture

The meta information of information chunks is structured as a "lean" conceptual model, ensuring that each information chunk is subsumed by a concept of a hierarchically organised model. Each server specialises in the implementation of a well defined set of these concepts. A new server can be integrated into the environment by defining new concepts, their attributes and behaviour. By treating concepts themselves as information chunks, the formal relationships between concepts can be made available (and partially extensible) at system runtime.

Internet technologies are especially relevant for inter-organisational co-operation and the virtual design office of the future. Internet technologies and standards, particularly related to email and the Web (SMTP, HTTP, HTML, CGI, 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 in ToCEE is especially designed to manage project member data, their work-lists, document databases, and product model browsing.

3.2 Concurrency

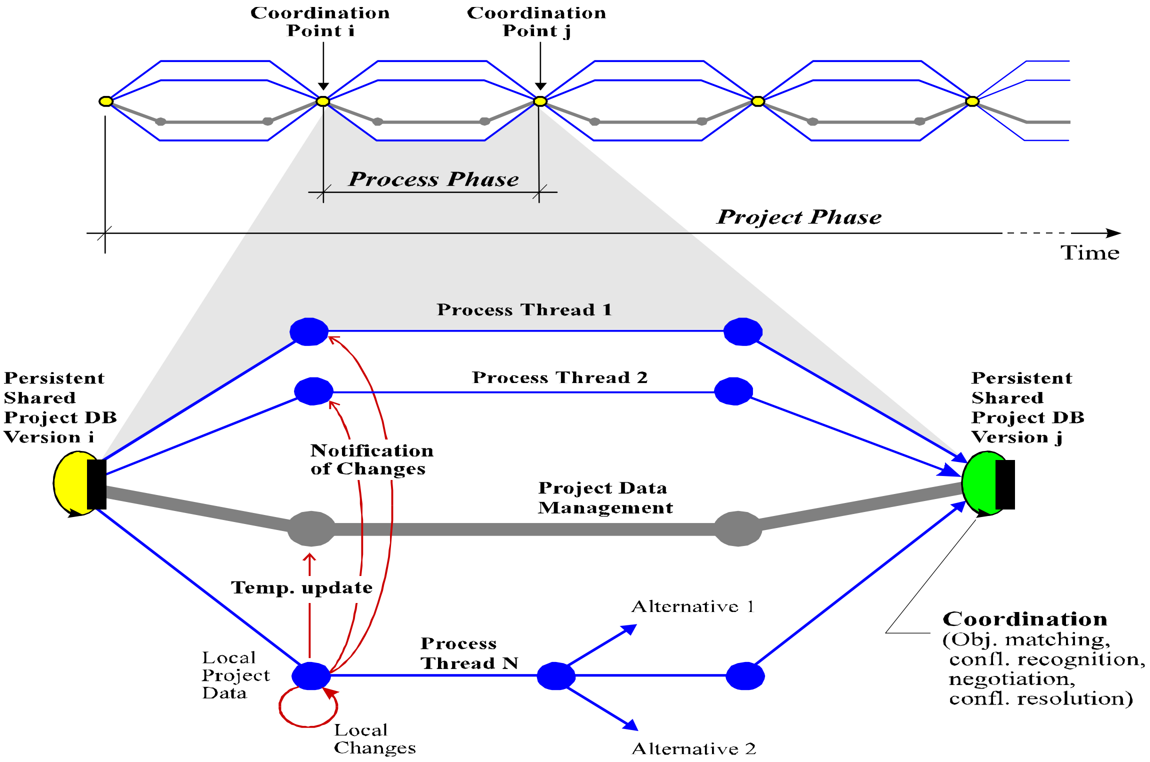


Fig. 4: Structured concurrent process progress

For concurrency in design, in construction, and in maintenance the information management, documentation, and archiving (product data) systems have to allow the existence of several versions of the information at the same time. This will be due to parallel working on the same product. The ToCEE objective is parallel independent working with a minimum of co-ordination and notification. As a result, conflicts will arise, which need appropriate methods to be handled.

A compromise between these two important objectives for concurrent and fast track engineering results in a project progress structured by co-ordination points. Between the co-ordination points, parallel independent work is allowed, where the individual participants, together with the project management, decide upon the needs and frequency of notification among the participants. Information logistics becomes therefore of utmost importance. Additionally, due to the growing inconsistency of data in the progress of parallel working, conflict management becomes an important management task. This task can be well supported by IT applying version management methods and applying artificial intelligence methods for conflict recognition and conflict propagation, i.e. propagation of the constraints due to conflicts and due to hypothetical solutions.

4. CONCLUSIONS

The goal of ToCEE is the development of an overall conceptual framework, software system prototype and discrete tools to support concurrent engineering. Such a system should handle various information types, including product and process information, document management, legal aspects and regulation processing. In co-ordinating the efforts related to management of this information, we have learned that an overall conceptual structure of a CEE system is required.

This paper documents one of the views on such a structure, which is based on the separation of the construction and the information system aspects. It includes a very abstract concept of an information chunk. The information chunk is the component which is handled by the information logistics system. To applications such a system provides a high level abstraction of a distributed, multi-server, multi-client environment.

The paper described the results achieved in the ToCEE project after one third of its course, and is clearly work in progress. The prototype environment under development will show to what extent the ides expressed are useful. This will be demonstrated through a scenario involving a large scale, real-life, engineering project covering many life-cycles of a project

5. ACKNOWLEDGEMENT

Research related to this paper has been carried out in the framework of the ESPRIT-ToCEE project which is financially supported by the European Union and the industrial partners. Their support is appreciated.

6. REFERENCES

Amor, R. and Clift, M. (1996) Document Models and Concurrent Engineering, in Turk, Z. (ed), Construction on the Information Highway, Proc. CIB-W78 Workshop, Bled'96.

Augenbroe, G. (ed) (1995) COMBINE 2 Final Report, EU/CEC Joule Programme, Project JOU2-CT92-0196, TU Delft.

Boehms, M. and Storer, G. (1994) ATLAS - Architecture, methodology and Tools for computer-integrated Large Scale engineering, in Proc. JSPE-IFIP WG 5.3 Workshop, DIISM'93, Tokyo.

Dean E.B., Unal R. (1992): *Elements of Designing for Cost*, Proc. AIAA 1992 Aerospace Design Conf., AIAA-92-1057, Irvine, CA.

Eastman, C. (1998). Conceptual Modeling of Buildings. Review, in P.Christiansson and H. Karlsson (eds) "Conceptual Modelling of Buildings", CIB Publication 126, Swedish Building Center, Solna, Sweden.

Gielingh, W. (1998). General AEC Reference Model (GARM), an aid for the intergration of applications specific data models, in P.Christiansson and H. Karlsson (eds) "Conceptual Modelling of Buildings", CIB Publication 126, Swedish Building Center, Solna, Sweden.

Henderson, L.R. (1991). "CALS as a solution for digital delivery of technical documents", Computer-Aided Design, vol.23, no.4.

IIC Consulting and Cimtech Limited: '1996 Engineering Document Management and Product Data Management Guide, An Introduction to EDMS and PDM and the Suppliers of Products and Services', ISBN 0-900458-71-2, 1996.

Luiten G., Froese T., Bj^rk B-C., Cooper G., Junge R., Karstila K., Oxman R. (1993). *An information reference model for architecture, engineering and construction*, in Mathur K.S., Betts M.P., Tham K.W. (eds) Management of information technology for construction, World Scientific Publishing, Singapore.

Scherer, R.J. and Sparacello, H.-M. (eds) (1996) COMBI Final Report, EU/CEC ESPRIT III Project No. 6609, TU Dresden.

Turk, Z. (1995). Document centered approach to design process modeling, R.J. Scherer (editor), Product and process modelling in the building industry, Balkema, Rotterdam, The Netherlands, 273-280.

Turner, J. (1998). A systems approach to the conceptual modelling of buildings, in P.Christiansson and H. Karlsson (eds) "Conceptual Modelling of Buildings", CIB Publication 126, Swedish Building Center, Solna, Sweden.

Watson, A. and Crowley, A. (1995) CIMsteel Integration Standards, in R.J.Scherer (ed) "Product and Process Modelling in the Building Industry", Proc. of ECPPM'94, Dresden, Balkema, pp 491-494.

Winner R.I., Pennell J.P, Bertrand H.E., Slusarezuk M.M.G. (1988): *The Role of Concurrent Engineering in Weapon Systems Acquisition*, Institute for Defense Analysis, IDA Report R-338, Alexandria, VA.