NEUTRAL OBJECT TREE SUPPORT FOR INTER-DISCIPLINE COMMUNICATION IN LARGE-SCALE CONSTRUCTION

SUBMITTED: June 2001.
REVISED: July 2001.
PUBLISHED: September 2001 at http://itcon.org/2001/3/
EDITOR: B-C Björk

Sander van Nederveen, Dr.
TNO Building and Construction Research, Delft, Netherlands
email: s.vannederveen@bouw.tno.nl

Frits Tolman , Prof.
Delft University of Technology, Netherlands
email: f.tolman@ct.tudelft.nl

*SUMMARY**:** Communication between disciplines in building and construction can be improved significantly by the proper use of Information and Communication Technology. For that reason, many research groups have been trying to achieve such improvement, especially by using Product Data Technology and STEP, and more recently the IAI Industry Foundation Classes. Unfortunately, the practical results of these efforts, in terms of tools that are used in practice, are still poor. The main reason for this seems to be that most research efforts have been too much top-down oriented, resulting in complex models and long-lasting developments. This paper presents an approach which tries to combine the sound ideas of Product Data Technology and STEP with a strong bottom up development strategy. The approach is based on Neutral Object Trees: hierarchical structures in which the objects that must be designed and built, are structured in a simple decomposition tree. The Neutral Object Tree approach has been used for the development of an Object Tree at the Dutch High Speed Railroad project, which is implemented in a PDM-system that is currently in use.*

*KEYWORDS**:** Object Trees, Communication in large-scale construction, Product Data Technology*

1. COMMUNICATION IN BUILDING AND CONSTRUCTION

The building and construction industry is facing great challenges. A large amount of work must be carried out, including a number of very large projects. In the Netherlands for example: the extension of Schiphol Airport, the second Maasvlakte, and railroad projects such as the Betuweroute and the HSL. In such large projects, but also smaller ones, high demands are put on the control of quality, time and cost.

Traditionally, the building and construction industry is characterized by dynamic partnerships between different disciplines from different organizations. In this situation, communication between different disciplines is a critical success factor. Therefore the aim for better control of quality, time and cost, often leads to an aim for better communication between disciplines.

In recent years a number of developments can be recognized that aim at better communication between disciplines, such as:

- New contract types,
- Classification and coding,
- Performance approach,
- Systems engineering approach.

1.1 New contract types

At the moment new contract types such as Design & Construct and Build - Operate - Transfer are very popular. The idea behind such developments is to make use of each party's capacities in an optimal way (De Ridder, 1994). For example by taking care that risks are managed by the party which is best equipped for the job.

1.2 Classification and coding

Classification is the distinction of (object) classes. Coding is the addition of codes to these classes. The best known example of classification and coding is the SfB-system. Traditionally, classification in building and construction used to aim at building elements. Later on, such classifications are extended with activity classes next to element classes (see for example the Dutch STABU system, see www.stabu.nl), with multiple decomposition levels, library structures, etc. In fact, developments such as these go beyond classification methods, and must be regarded as building product modelling developments. An example of this is the work of Working Group 6 of ISO TC 59/SC 13 (ISO TC 59, 2000).

1.3 Performance approach

The essence of the performance approach is that the objectives of a (building) project are formulated in terms of quantifiable performance requirements, not in terms of prescribed solutions.

1.4 Systems engineering approach

The systems engineering approach means that a product is seen as a collection of systems that should take care of a certain performance (INCOSE, 1998). For example: a space system, a structural system, a heating system.

Developments aiming at better communication such as described above, are closely related. For example: in a Design & Construct project, a key role is played by performance-based specifications; and it is often worthwhile to do this by using systems requirements.

2. INFORMATION AND COMMUNICATION TECHNOLOGY IN BUILDING AND CONSTRUCTION

In communication in building and construction, information and communication technology (ICT) of course plays a key role. But when the state of the art of ICT in building and construction is considered, then it must be concluded that ICT in building and construction practice is still on a rather low level. For example, ICT in design is mainly based on CAD-systems (drawing systems) and exchange of CAD data. Only on a modest level, some integration exists between CAD systems and for example CAE-systems (calculation programs).

This low level state of the art was already recognized in the mid eighties, leading to the initiative to the well-known ISO-STEP standard (officially ISO 10303) (ISO, 1993). The STEP standardization initiative aims at electronic communication of product information on a semantic level, based on standardized product models, i.e. standardized information models of product data to be exchanged between participants in engineering environments.

Since the start of STEP in the mid eighties, many follow-up initiatives and projects have started. But the results of all these efforts are rather disappointing for the building and construction industry. For other industries, specific STEP-based standards have been developed, for example for the process industry (STEP AP 221 and the EPISTLE work), for shipbuilding and for the automotive industry.

But for building and construction a widely accepted standard based on STEP is still missing. This is probably due to the following problems:

- The building and construction sector is fragmented, often small-scale, nationally oriented, and without dominant parties; therefore it is difficult to reach agreement on sector-specific standards.
- The ICT-sector on the other hand, is internationally oriented; therefore it is even more difficult to achieve ICT-support for sector-specific standards.
- Many different ICT-approaches for support of standards exist, and new approaches emerge almost continuously.

A recent development in the area of electronic communication for building and construction is the development of Industry Foundation Classes (IFCs) by the International Alliance for Interoperability (IAI, 1997). These IFCs are essentially standardized CAD objects that contain both geometric and semantic product data. The IAI that is developing the IFCs, is an international consortium of CAD vendors, such as AutoDesk, Bentley, Nemetschek, and many other parties. The IFC development has some important advantages above STEP, for example the leading role of the software vendors. But also the IFC development is taking place very slowly.

3. TOWARDS BETTER COMMUNICATION AND INFORMATION EXCHANGE

Besides STEP and IFCs there are a few important developments that may help to achieve practical solutions for electronic communication in building and construction.

3.1 STEP and OO

One of the problems of the STEP work is its slowness, especially compared to developments in ICT. As a result, several initiatives have started aiming at the application of the latest technologies for STEP work.

One example of this is the support of behaviour using the object-oriented CORBA technology (OMG/CORBA, 1998). In the European project VEGA work is done on the application of STEP combined with CORBA in order to support workflow management in industries such as building and construction (VEGA, 1998).

3.2 Minimal Models

Another trend can be called "the minimal approach". According to this approach data models are kept very small in order to achieve a simple format for data exchange in building and construction. Elaborations of this approach are e.g. aimed at building geometry (Tarandi, 1998), and on integration with Electronic Data Interchange (EDI) (De Vries, 1996).

3.3 View models

With respect to new concepts for communication the so-called view-approach is important. The view approach, or more precisely the discipline view approach, starts from the observation that different participants in building and construction represent different disciplines, each of which has its own view on design information (Van Nederveen, 1993). As a result, each participant has its own specific information requirements, which must be supported by specific information models (so-called view models). For communication in building this means that support of view conversion is a first prerequisite.

The view approach is elaborated in different ways. For building and construction the approach is worked out most extensively in the European project ATLAS (ATLAS, 1993). As shown in this project, the big practical issue in the view approach is that the relationships between the various view-specific information models (the view conversion) become too complex, leading to too costly implementation and maintenance of conversion software.

3.4 Internet and XML

A very promising trend in the context of basic technologies, is the ongoing development of Internet technology, especially with XML. However, work in this area is not yet matured. An interesting project aimed at E-Commerce in building and construction using XML is the EU-project eConstruct (Tolman & Böhms, 2000).

3.5 Evaluation

Despite all the work done within and outside STEP, we must conclude that in building and construction very little results have been achieved in terms of tools used in daily practice.

Because of the specific characteristics of building and construction, it seems that the introduction of a new concept for electronic communication on a semantic level can only be successful when a very pragmatic and bottom-up approach is used. At least much more bottom-up than in the various projects in the past.

4. NEUTRAL OBJECT TREES

In order to achieve an approach that is pragmatic, bottom-up, allowing fast implementations and quick returns of investment, it was necessary to reconsider a number of "STEP-habits", and ask ourselves questions such as: "Do we really need complex data models? What kind of relationships and structures do we really need? Can we avoid the difficult subject of geometric models and shape description?". But most importantly, we started really bottom-up, with the development of a model of a single project, without having a reference model or type-model in advance.

The experiences of this work led to the so-called *Neutral Object Tree* approach, of which the main characteristics are listed below:

- An Object Tree is an instance model, it describes the objects of one particular project.
- A Neutral Object Tree is independent of software vendors, building participants and standardization developments, etc.
- Objects are function performers.
- An Object Tree is a decomposition tree (using: contains, or consists of).
- An Object Tree supports a minimum set of relationships.
- For shape description a reference to CAD drawings, or a VRML shape description is specified.
- An Object Tree can be made largely by hand, using cut and paste (think of it as a simple hierarchy as used in the Internet Explorer).

4.1 An Object Tree is an instance model

Or in common English: an Object Tree describes a single thing, not a class of things. The White House is an example.

One of the "STEP habits" stated above is to start with a product type model (or reference model, or STEP ARM), a model that describes a certain class of objects, such as buildings, roads or viaducts. Type models have to undergo a standardization process. Once a standard is available ICT vendors have to implement it in their tools. Next Building-Construction companies have to buy these systems. And finally a consortium has to agree on using the standard and tools in a project. A long long way.

Instead, the Object Tree approach starts right away with the creation of the instance model that supports electronic communication. True, the Neutral Object Tree will not be as elaborate as the future product models, but at least we don't have to wait for another decade.

4.2 A Neutral Object Tree is independent of software vendors, participants and standards

This characteristic follows from risks such as being dependent on vendors that might change their strategy, or just might vanish. Or being dependent on organizations that might change (narrow) their strategy and stop their commitment. Or being dependent on slow acting standardization committees with long-lasting procedures. An Object Tree can be created in a matter of days. The real problem is getting everybody on the same line, but that is a problem anyway. The bottom-up approach at least starts at the right place: the bottom.

4.3 Objects are function performers

The "building blocks" of the Object Tree are the objects. Objects are regarded as physical things that perform a function. This function can be the realization of a required performance, as specified in a Requirements Specification.

This means that a Neutral Object Tree can be regarded as a solution tree, in which also the functions are specified for which the objects are solutions.

4.4 A Neutral Object Tree is a decomposition tree

As the term "tree" already suggests, Object Trees have a hierarchical structure. Now there are many ways in which a hierarchy can be made in a product structure. Nowadays object hierarchies, or trees are usually specialisation trees (following the Object Oriented approach). But the Neutral Object Tree approach aims at simplicity, and therefore it proposes to use one hierarchical principle: decomposition ("consists of").

However, decomposition can still be applied in different ways. For the Object Tree two decomposition principles are used:

- Subsystem decomposition, in which an assembly is decomposed into groups of objects that share a location,
- Aspect system decomposition, in which an assembly is decomposed into groups of objects that share a specific aspect or role.

Both decomposition principles are needed and are therefore part of the Object Tree approach. But the relationship between elements of these decomposition trees can be very complex. Therefore this relationship is not modelled explicitly in the Object Tree.

4.5 An Object Tree contains a minimum set of relationships

Relationships (such as “is connected to”) between objects can easily lead to a very complex model structure. For that reason a Neutral Object Tree contains only a minimum set of relationship types. First of all, there are the decomposition relationships as described above. Furthermore the only other relationship type is the physical interface. No other functional, logical or any other kind of relationships is used.

4.6 For shape description a reference to CAD drawings or VRML models is specified

For shape description a number of methods exist, but again these methods can easily lead to very complicated information models. Once more, a simple solution is chosen: the use of references to CAD drawings. In other words, if a user wants to know about the shape and dimensions of an object, then he should find a function that brings him to the drawing in which he can find what he is looking for. An alternative is to represent each object as a local VRML model and let the browser do its work. In the future Neutral Object Trees will be made viewable over the Internet using VRML, Java3D or X3D.

4.7 A meta-model for Object Trees

Below the characteristics discussed above are modelled in an EXPRESS-G diagram, see Fig. 1.

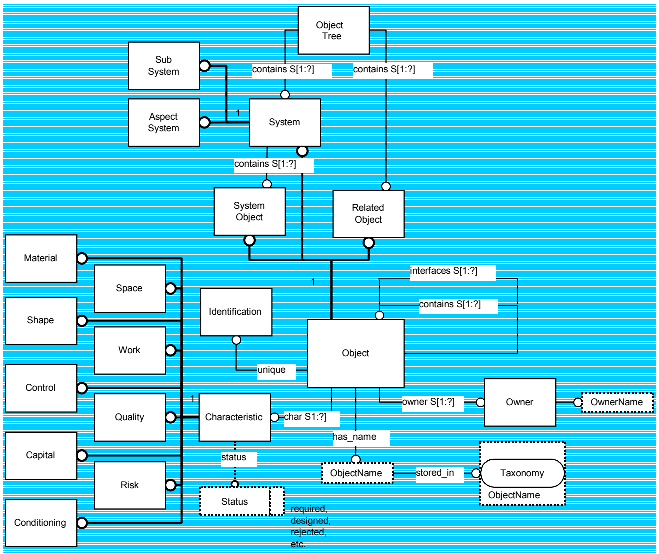

FIG. 1: The Object Tree Meta Model

The main constructs in the Object Tree Meta Model are the following:

- An Object Tree (at the top) contains Systems, which in turn contain System Objects. Furthermore an Object Tree contains Related Objects (relevant Objects in the environment of the System Objects).
- System Object and Related Object are both subclasses of the generic class Object.
- The generic class Object (in the center) has a number of generic properties and relationships, including:
  - An Object has an ObjectName that can be stored in a Taxonomy,
  - An Object has a unique Identification,
  - An Object has Characteristics, which in turn may have a Status,
  - An Object may contain other Objects (decomposition),
  - An Objects interfaces other Objects (connectivity),
  - An Object has one or more Owners (actors responsible for the Object).

5. IMPLEMENTATION OF OBJECT TREES

As the Object Tree is basically rather simple, the implementation of Neutral Object Trees does not have to be very difficult either. In fact it is even possible with systems such as Excel or Access, but such systems fall short in support of either data management and maintenance (Excel) or user interface (Access). A better solution is to pick one of the commercially available Product Data Management (PDM) systems, and to tailor the system according to the specific needs of the organization.

In the near future new operating systems and Internet software will further enhance the possibilities for implementation of Object Trees.

6. THE HSL CASE

The Dutch High Speed Line (HSL) project aims at the design and realization of the Dutch part of the new railway track for high speed trains between Amsterdam and Paris. In this project the approach described above has been applied in the so-called HSL Object Tree. This has been done by a development team at the HSL Project Organization. The first author was one of the members of this team between 1997 and 1999.

The HSL Object Tree development has resulted in a decomposition structure with of course the entire HSL track as top of the tree. The HSL track decomposes in a few steps into some thousands of HSL objects such as bridges, viaducts, tunnels, noise reduction screens, cables and ducts etc., see Fig. 2.

In fact the HSL approach has been even simpler than the approach advocated in this paper. For example the functional side of objects, the aspect systems decomposition and the definition of physical interfaces are only elaborated in part. The implementation of the HSL Object Tree has been done using Excel, Access, and the PDM system SmarTeam subsequently.

The SmarTeam system was also used for the management of CAD drawings, which was integrated with the HSL Object Tree. I.e. CAD drawings could be accessed by navigation through the Object Tree and selection of the object that is displayed on the drawing.

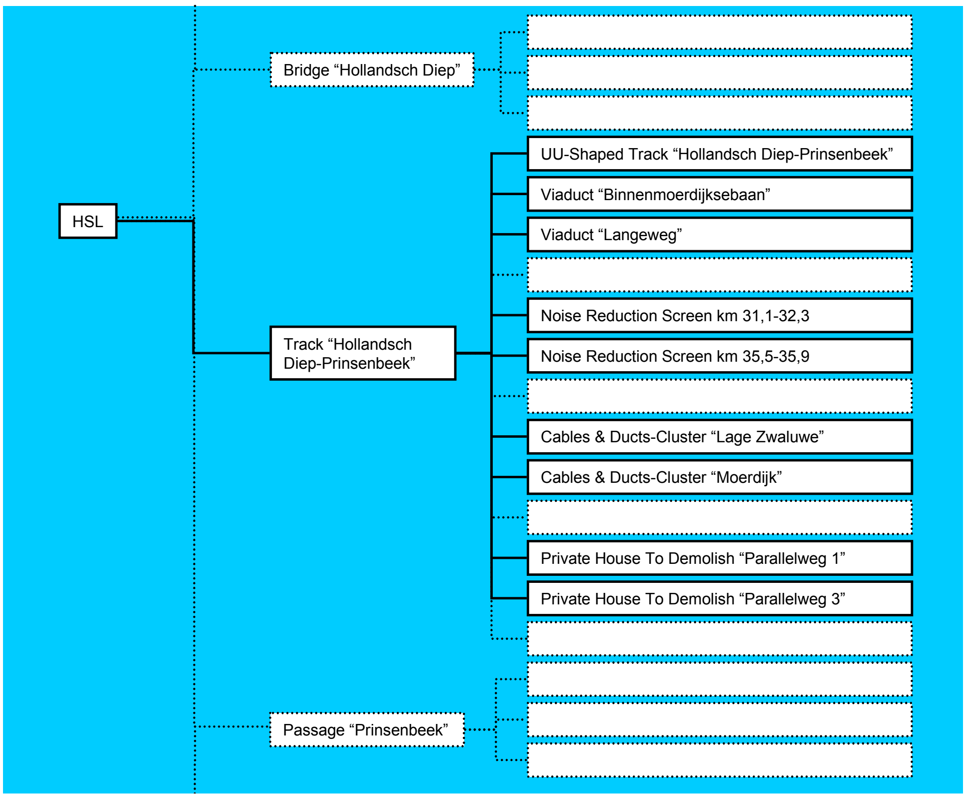

FIG. 2: A sample of the HSL Object Tree

Fig. 3 shows a screendump of the SmarTeam implementation.

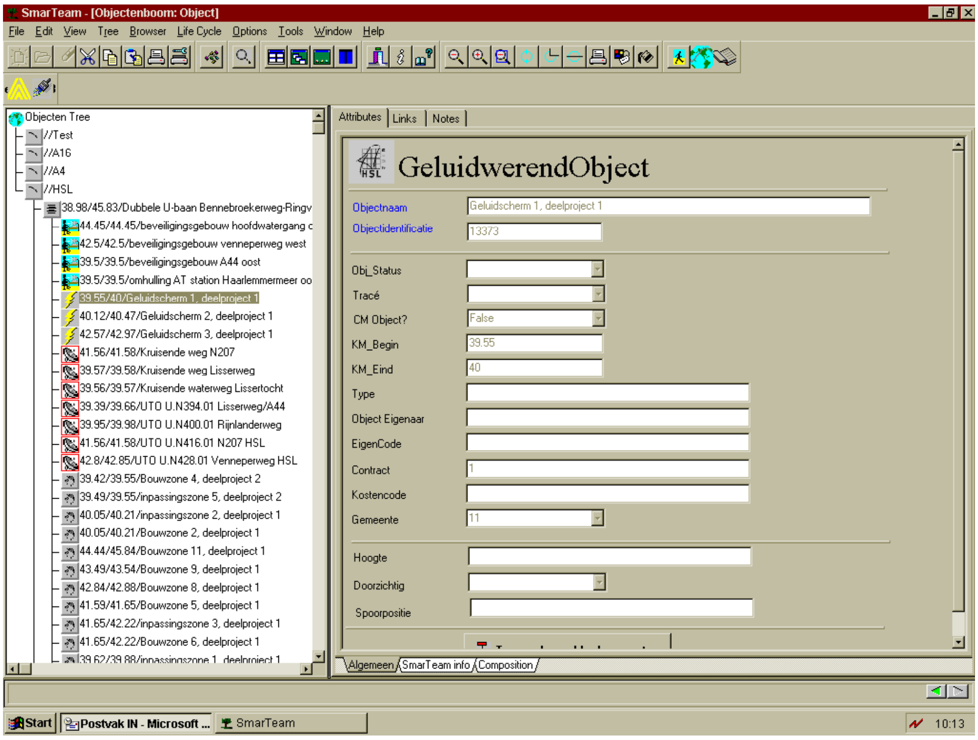

FIG. 3: Implementation of the HSL Object Tree in SmarTeam

The HSL Object Tree has been a very good opportunity to find out the feasibility of Object Trees in practice. The HSL project has been a typical example of a project in which product data are developed in a natural, unstructured, unorganized, anarchic way. Product data tend to emerge in many kinds of lists and spreadsheets and alike, originating from the many different design disciplines in the project.

So a first observation at the HSL is that it takes a lot of effort, time and patience to obtain a single Object Tree that fulfils the basic requirements of all design disciplines – a lot more than one would expect from an R & D point of view. Secondly, and strongly related to this: the saying "Keep It Simple & Stupid" is very true in this kind of circumstances: lots of data, many participants, and an informal, creative and ad hoc culture.

But when these points of attention are kept in mind, the HSL experience shows that Object Trees can play a valuable role in projects such as the HSL:

- It adds necessary structure to the product data being produced,
- It enables standardization of object names and properties such as location,
- It provides a checklist for various purposes,
- It provides a basis for other purposes such as the management of CAD data,
- And last but not least, it provides a basis for interface management and risk management.

But the ultimate effectiveness of Object Trees is largely dependent on the acceptation and commitment of the management.

On the implementation side, the following conclusions can be drawn:

- For basic implementations and prototyping work, general purpose software such as spreadsheets and databases such as Excel and Access, is useful. A great advantage of such software is its low threshold: it is available for everyone, and known by almost everyone. However, this kind of software soon falls short in terms of user-friendliness and maintainability.
- For more serious implementations, commercially available PDM-systems can do the job (and do not have to be very expensive). A successful PDM-implementation has been carried out at the HSL project using SmarTeam.
- For future projects, Internet/intranet based solutions are very promising. Such solutions may drastically simplify issues on concurrent design, such as working at different locations, exchange of shape information (using VRML or so), etc.
- Alternatively, it might be worthwhile to try and combine the Object Tree approach with other PDT-approaches. For example: to use parts or subsets of IFCs, EPISTLE or the LexiCon (Woestenenk, 2000). However, there is a significant danger that such a combined approach will lead to modelling complications and, consequently, to huge delay.

As a bottom line, the HSL-experience has proven that Object Trees can definitely be feasible in practice. Compared to other PDT work, the Object Tree approach has proven to work, while this is doubtful in many other PDT efforts.

7. CONCLUSIONS

In this paper the application of Neutral Object Trees as a common basis for meaningful electronic communication in large scale building and construction projects is discussed. An Object Tree is essentially a decomposition (part-of) hierarchy that forms the major breakdown of the constructed facility. As such, Object Trees are primarily aimed as an aid for engineering management (for example for interface management), rather than for the support of creative design.

The main conclusions of the research are:

1) For improvement of communication in building and construction the semantic representation and exchange of product data is a prerequisite.

2) The development of methods and tools for better communication in building and construction easily fails due to: the complexity of the information models, unfortunate ICT choices, and disappointing developments in standardization.

3) By using Neutral Object Trees a simple yet complete information model for design and engineering can be developed in a short period of time, which can serve as a basis for better communication.

4) Such a Neutral Object Tree must meet the following requirements:

- The Object Tree must be developed bottom-up, i.e. objects (instances) first, classes later.
- The Neutral Object Tree must be *neutral*, that is independent of software vendors, participants and standardization developments.
- Objects must be seen as physical things that perform a function.
- The Object Tree must be a decomposition structure, with both subsystems decomposition (shape driven) and aspect systems decomposition (aspect driven); the Object Tree should not have any other hierarchical structure.
- The Object Tree must support decomposition relationships as described above, furthermore physical interface relationships, but no other types of relationships.
- Shape description must be taken care of by a reference to CAD drawings, or by simple local VRML shapes.

Such an Object Tree can be developed in a relatively short time. Moreover it can also be implemented in a short time, for example using commercially available PDM systems.

5) The Object Tree can be elaborated further as follows:

- (Further) Development of classification and standardization of object names and object types.
- (Further) Development of libraries of standard objects, but also of standardized resources and processes.
- (Further) Development of management methods using the Object Tree, for example interface management and risk management.
- Generalization of the Object Tree towards a type model to support PDT developments.

8. ACKNOWLEDGEMENTS

The research presented in this paper is mainly carried out in a PhD-research project that has led to the first author's thesis (Van Nederveen, 2000). The research has been sponsored by the Dutch Technology Foundation (STW) and TNO Building and Construction Research.

9. REFERENCES

ATLAS (1993). ATLAS consortium, *ATLAS Public Project Overview*.

De Ridder H.A.J. (1994). *Design and Construct of Complex Civil Engineering Systems, A new approach to organization and contracts,* Delft University Press, ISBN 90-407-1027-9.

De Vries B. (1996). "*Communication in the Building Industry*", PhD-thesis, Eindhoven University, ISBN 90-6814-538-X.

IAI, International Alliance for Interoperability (1997). "Industry Foundation Classes", Release 1.5. Website http://iaiweb.lbl.gov.

INCOSE, International Council on Systems Engineering (1998). *"Systems Engineering Handbook"*, January. Website: http://www.incose.org.

ISO/TC184 (1993). “Part 1: Overview and fundamental principles,” in *Industrial automation systems and integration – Product data representation and exchange*, International Standard, edition First edition 1994-12-15, ISO, Geneva, Switzerland, ISO 10303-1:1994(E).

ISO/TC59 (2000). ISO PAS 12006. “Building construction — Organization of information about construction works — Part 3: Framework for object-oriented information exchange”.

OMG/CORBA (1998). “The Common Object Request Broker Architecture (CORBA) Specification”, Revision 2.2, www.omg.org/techprocess/meetings/schedule/Technology_Adoptions.html, March.

Tarandi V. (1998). Neutral Intelligent CAD Communication (information exchange in construction based upon a minimal schema), PhD thesis, KTH, Sweden. Also downloadable from http://www.recm.kth.se/cme.htm.

Tolman F. and M. Böhms (2000). “Electronic Business in the Building and Construction Industry: Preparing for the New Internet”, *Proceedings CIT 2000/CIB W78,* Reykjavik.

Van Nederveen S. (1993). S. van Nederveen, "View integration in Building Design", in proceedings *Management of Information Technology for Construction*, in Singapore, eds. K.S. Mathur, M.P. Betts, and K. Wai Tham, pp. 391-406, World Scientific Publishing Co. Pte. Ltd (CIB W78).

Van Nederveen S. (2000). *Object Trees – Improving Electronic Communication between Participants of Different Disciplines in large-scale Construction Projects,* PhD-thesis, Delft University, ISBN 9013619-3.

VEGA (1998). J. Stephens, M. Böhms, M. Köthe, J. Ranges, R. Steinmann, R. Junge, A. Zarli. “Virtual Enterprise using Groupware tools and Distributed Architectures”, in: R. Amor (ed), *Product and Process Modelling in the Building Industry,* British Research Establishment, Watford, UK.

Woestenenk (2000). “The LexiCon: An update”. In R. Goncalvez, A. Steiger-Garcao and R. Scherer, *Product and Process Modelling in Building and Construction,* Balkema, Rotterdam.