# Object Trees

Improving Electronic Communication between Participants of Different Disciplines in large-scale Construction Projects

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,
op maandag 17 april 2000 te 16.00 uur

door

Gilles Alexander VAN NEDERVEEN

bouwkundig ingenieur
geboren te Hengelo (O)

The research presented in this thesis is sponsored by the Technology Foundation (STW) and TNO Building and Construction Research

Published and distributed by:

Sander van Nederveen
Julianalaan 114
2628 BK Delft
The Netherlands
Phone: +31 (0)15 2131768
E-mail: s.vannederveen@bouw.tno.nl

ISBN 90-9013619-3

# Stellingen

bij het proefschrift:

## Object Trees

Improving Electronic Communication
between Participants of Different
Disciplines in large-scale
Construction Projects

*Sander van Nederveen*

1. De 30 % productiviteitsverbetering in het bouwproces die volgens Latham [1994] en anderen haalbaar is, kan voor een belangrijk deel (orde van grootte 3-5% op korte termijn) bereikt worden door inzet van elektronische communicatie op basis van betekenisvolle informatie [Wix & Liebich, 1998].

2. Een neutrale objectenboom (neutraal = onafhankelijk van software-leveranciers, participanten in het bouwproces, standaardisatie-ontwikkelingen, etc.) is de eenvoudigste vorm van een productmodel, met behoud van de positieve kenmerken zoals het ondersteunen van betekenisvolle communicatie en hergebruik van technische oplossingen en kennis, en zonder de veel voorkomende negatieve kenmerken als complexe modellen en een lange ontwikkelingstijd.

3. Doordat de Object Tree-benadering een veel kortere ontwikkelingstijd mogelijk maakt dan andere PDT-benaderingen, is ze voor snel veranderende organisaties (waar snelle systeemontwikkeling een eerste vereiste is) wellicht de enige haalbare mogelijkheid om elektronische communicatie op basis van betekenisvolle informatie in te voeren.

4. De Object Tree-benadering kan worden gebruikt als uitgangspunt voor standaardisatie door eerst een aantal Object Trees te ontwikkelen, en deze vervolgens te generaliseren tot een type-model voor bijvoorbeeld lijninfrastructuur. Een dergelijk bottom-up-traject heeft het grote voordeel ten opzichte van het top-down-traject van ontwikkelingen als STEP, dat de praktijk kan beschikken over werkende hulpmiddelen voordat consensus is bereikt over een standaard.

5. Gebrekkige communicatie in grote bouwprojecten komt voor een belangrijk deel doordat betrokkenen vasthouden aan een (in kleine projecten vaak uiterst succesvolle) informele werkwijze.

6. Als de overheid wil scoren met nieuwe aanbestedingsvormen als Design & Construct, zal zij beter moeten leren specificeren

7. Zelf meedraaien in een praktijkproject leert veel meer over de behoeften van de bouwpraktijk dan samenwerken met vertegenwoordigers uit de praktijk in een onderzoeksproject.

8. Ook voor het verbeteren van elektronische communicatie in de bouw geldt de uitspraak van Einstein: zo simpel mogelijk, maar niet simpeler.

9. Het gaat bij informatiemodellen niet om "wat is waar", maar om "wat is handig" (naar Poincaré).

10. Verbetering van de mobiliteit voor iedereen is meer gebaat met een Randstad-metro dan met een HSL.

11. Een werkgever die veel tijdelijke contracten aanbiedt, ontmoedigt zijn werknemers om dicht bij zijn werk te gaan wonen en draagt aldus bij aan het fileprobleem.

12. Wie zelden met de trein reist, heeft relatief vaak last van vertraging.

# Preface

This thesis presents the results of a study into computer supported information management of large-scale (international) building and construction projects. The total amount of information required by, and produced in, large-scale building and construction projects, is gigantic and tends to become even more so in the years to come. It is fair to say that all this information is much too much to be managed by an individual or even by a small team and, in general, information management in large-scale building and construction projects is poor. Results of insufficient information management and miscommunication can be seen daily in every project and on every site.

In theory, Information and Communication Technologies can help to solve part of the problem. But before ICT can start *helping*, it first adds to the problem. The traditional paper based Information System of the Building and Construction industry is now partially extended with a second, electronic Information System that partly holds the same information and partly holds different information, thus creating additional information bottlenecks.

This thesis presents the results of a study into the question how the Building and Construction industry can improve its project information management, and tries to formulate a number of recommendations for the future.

The study focuses on the requirements of large-scale building and construction projects, because there the needs are felt most, and the power to do something about it exists.

This thesis finalizes a period of more than nine years of Ph.D.-research. During this period I was employed by the department of Information Technology for Building and Construction[1] of TNO Building and Construction Research in the Netherlands. Over the years I have been involved with very different projects. In the early nineties, I worked on STEP-related EU-projects such as ATLAS and COMBINE. From a scientific point of view, these were very interesting projects, in which we had the chance to explore issues on building product modelling and view integration in depth. But the projects had little practical results, in terms of tools that proved to be successful in building practice.

In the last three years, I worked on the HSL[2] Railroad project, thus in building design/engineering practice. The difference was a lot bigger than I had expected. In the HSL project there was a need for many things, but definitely not for the STEP-related work that I was familiar with. On the other hand, there was indeed a need for management of simple object data, resulting in the initiative of the HSL Object Tree. To me, the work on the HSL Object Tree learned a lot about how to start with Product Data Technology (PDT) in practice.

During these nine years I have been working on this thesis. To me it feels that for all of these years I have been chasing a running target which most of the time ran faster than I did. About three years ago, I was at a point that was interesting from a scientific point of view – but for which hardly anyone showed interest. Inspired by the HSL experience, this research became a lot more pragmatic and practice-oriented.

As a result, this thesis does not introduce revolutionary concepts for PDT. In fact the used technology is rather mainstream, though the basic concepts are still the same. But this thesis does not aim at revolutionary concepts. This thesis aims at an approach that can easily be applied in practice. And as such, I hope that this research can help to make the gap between PDT-research and practice a little narrower.

The research was sponsored by the Dutch Technology Foundation (STW), for which I feel grateful. The Foundation sponsored the Computer Integrated Construction (CIC) project (DCT99.1891) where six Ph.D. students worked on different but related aspects of Product Data Technology for Civil Engineering applications.

[1] Formerly: the department of Computer Integrated Construction

[2] Hign Speed Line

Many people have helped me along the way, for which I feel very grateful. First of all, Frits Tolman, who kept believing in me and guided me in the right direction whenever needed. Secondly, the reviewers of the draft, who provided many valuable comments: Martin Lamers, Bart Luiten, Michel Böhms, Theo van Rijn, Peter Willems and Johan Neuteboom and the members of the commission. And I would like to thank my Ph.D. colleague researchers in the CIC-project and the other current and former colleagues at TNO for their team spirit and help. Also many thanks to the HSL-people for letting me find my way in "the real world".

And finally, I would like to thank my wife Gea. I should never call her impatient again.

Delft, February 2000,

Sander van Nederveen

# Table of Contents

# Introducing the Problem

*This chapter introduces the subject and describes the scope of the research.*

## 1.1 Introduction

The Building and Construction industry, at least in Europe, is facing strong demands to increase its efficiency and effectiveness. Clients and facility operators demand better quality, faster and cheaper built facilities incorporating more complex technology. The environment is getting extremely vulnerable and large areas are overcrowded and filled with infrastructural works; consequently, governments have considerably increased the regulatory constraints on safety, waste, durability and energy consumption.

In an effort to cope with the increasing demands, the Building and Construction industry – like every other industry – started to use Information and Communication Technologies (ICT). Computers and computer networks nowadays play an important role in the design, engineering and realization of large-scale building and construction projects. Most professionals are no longer able to do their jobs without using some sort of Computer Aided system. Especially the engineering disciplines (Structural, Mechanical, HVAC, Electrical, etc) are heavily relying on computers for analysis and simulation.

With the arrival of computers however, the Building and Construction industry faces a new kind of problem, i.e. an increased fragmentation due to insufficient electronic communication. As the industry already suffers the consequences of severe fragmentation, additional bottlenecks are undesirable.

Several disciplines early on recognized the problem and started to look for solutions. However, efforts to provide meaningful[3] electronic communication between project participants belonging to *different* disciplines (subsequently called *interdisciplinary* communication) in the Building and Construction industry are scarce and up till now largely unsuccessful.

The current state of the art of electronic[4] interdisciplinary communication is analysed in the next sections. The reasons for the current lack of achievement are being discussed as well.

While looking into the existing efforts to develop an electronic Information System for the Building and Construction industry, at first the problem proved even worse. There is not *one* development going on, but several, largely conflicting developments. Each development with its own merits and demerits, each with its supporters, each costing a lot of money and each providing its own small part of 'an' electronic Information System, and each adding to the problem and the solution.

## 1.2 The RS-IS Paradigm

In order to describe the position and role of information systems in building and construction, the so-called RS-IS-paradigm is used [Brussaard 1980] [Sol 1988].

Following the RS-IS-paradigm, building and construction processes can be seen as transformation processes in which inputs such as construction materials and products are transformed into artefacts. The material transformation process is called the Real System (RS).

Complex transformation processes such as construction processes are always supported by another transformation process where information about requirements and constraints is transformed into information about the actual state of the artefact. The second transformation system is called the Information System (IS).

[3] Meaningless communication, i.e. strings of bits and bytes, can always be communicated.

[4] The word 'electronic' will be omitted in the rest of this text, because this study focuses on ICT, which is by definition electronic.

Figure 1.1 shows both systems and their interactions. The representation is called the RS-IS paradigm.

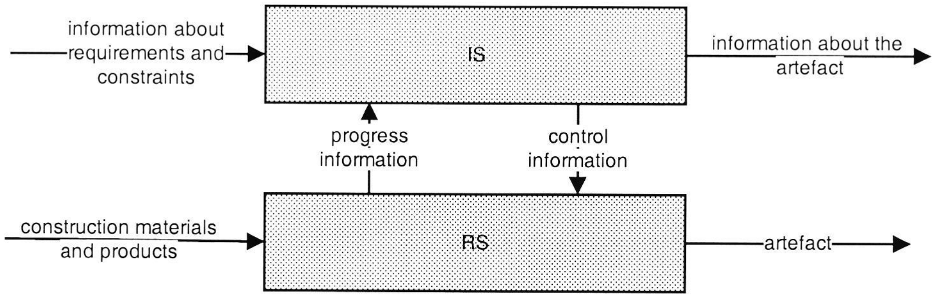

information about requirements and constraints
IS
information about the artefact
progress information
control information
construction materials and products
RS
artefact

*Fig 1.1. Real Systems (RS) transform construction materials and construction products into artefacts. Information Systems (IS) transform information about requirements and constraints into information about the artefacts. The IS sends control information to the RS and receives progress information from the RS.*

In order to fulfil its role the IS uses a *model* of the RS. To be effective, this model, like all models, should of course contain the essential characteristics of the construction processes and the actual state or states of the artefact.

Also important is the way the model is *represented*. If the information logistics is poor (not getting the right information, in the right format, at the right time, in the right place), as often is the case, numerous things can go wrong.

Traditionally the IS in the Building and Construction industry is based on documents, until recently mainly on *paper* based documents like technical drawings, schedules, reports, regulations, charts, diagrams, and such. Not only does the IS contain lots and lots of documents, it also holds data *about* these documents (meta-data). Meta-data describes how to develop a technical drawing, how to do calculations, or how to apply codes and regulations. In this process usually a large number of people is involved, each with its own skill and potential to change.

This brings us to a typical characteristic of the building process: its multi-disciplinary character. Many participants from different disciplines work together, exchange information but have their own view on the project. In this environment co-ordination and interdisciplinary communication are critical success factors.

With the appearance of the computer there came an alternative to the paper based IS which partly was realized quite soon, and partly will not be realized for another couple of decades.

This is the setting of the research. An industry that is truly fragmented is facing increasing demands for more efficiency and effectiveness. One of its efforts to meet the challenge, i.e. increase the usage of ICT, is partly successful, but partly contributes to the fragmentation. How to proceed from here?

## 1.3 Research Question

Loosely and somewhat abstractly formulated the research question is "How to use state of the art Information and Communication Technology, to improve interdisciplinary communication in large-scale building and construction projects?"

## 1.4 Structure of the Thesis

Chapters 2 to 5 deal with the analysis of the problem. Trends in building practice are examined and compared to other industries, developments in ICT are analysed, and research efforts on communication in building and construction are reviewed. From the analysis chapters the conclusion is drawn that the various efforts have led to very little results that work in building practice. Also some clues are given why this is the case.

The conclusions of the analysis lead to a reformulation of the research question in chapter 6. The research question is answered in chapter 7, in which a new approach for interdisciplinary communication is presented.

Chapter 8 discusses a case study where part of the theory developed in the previous chapters is put into practice. Chapter 9 evaluates the pros and cons of the proposed solution following from the limited case experience. Chapter 10 presents the final conclusions and some recommendations for the future.

# 2 Interdisciplinary Communication

*This chapter analyses earlier attempts to improve the competitiveness of the Building and Construction Industry by increasing its potential to communicate between disciplines.*

## 2.1 Introduction

Communication between disciplines in large-scale building and construction projects, other than an occasional email, is currently not really supported by ICT. The main reasons are: the fragmentation of the industry, the one-of-a-kind character of each new project, which is also performed by a new team, the traditional sequential project organization with its rigid division of responsibilities, and the low-tech nature of the construction processes and the construction people. Until recently there were not many incentives for the industry to increase its competitiveness. The last ten years however the situation has been changing rapidly.

## 2.2 Improving Communication

Applying Information and Communication Technology is of course not the only way to improve the industry's competitiveness and communication. Improvements have, among others, been made by:

- Improving project cultures,

- Adopting new contract types,
- Applying common classification and specification systems,
- Applying performance concepts,
- Applying systems decomposition.

In the next sections these efforts will be discussed in more depth.

### *2.2.1 Changing the Project Cultures*

Traditionally the project culture in Building and Construction projects is not an example of mutual trust and generosity, on the contrary. However, the last decades important strides have been made in areas of improved project culture, including:

- *Partnering* (forming regular working relationships between organizations transcending individual projects, common in other industry settings and other countries like Japan),
- *Mutuality* (injecting better common purpose and mission into projects, sharing problem ownership, and creating trust), and
- *Ability* (construct-ability, maintain-ability, use-ability) with upstream involvement of the "owners" of downstream knowledge, rather than just-in-time involvement.

In other industries, especially Automotive, changing the business culture has brought the concept of Lean Production, which is mainly based on trust and profit sharing. In our industry Lean Construction is still mainly a research subject.

### *2.2.2 Adopting New Contract Types*

New contract types like Design-Built and Design-Built-Operate assume improved communication resulting from the fact that one company, or one consortium, is responsible for the design *and* realization of the project, or even for the proper functioning of the resulting artefact.

At the moment Design-Built and Design-Built-Operate contracts are very popular, and in many occasions improvements in time, cost and quality have been satisfactorily realized, relieving many local and national government bodies of their worries. Another type of contract, the Privately Funded Initiative, is a next step along this road, a step that probably is even more attractive to government clients, because now also the financing scheme is taken over.

Whether this type of contracts really improve communication is not quite sure, as discussed in De Ridder [1994], who concludes that contracts alone are only one part of the solution.

### *2.2.3 Applying Common Classification and Specification Systems*

Classification and coding of building and construction information has been recognized as one of the most important technologies for improved interdisciplinary communication. Already fifty years ago the Swedish classification system SfB was developed and soon was starting to spread throughout Europe.

The initial objective of SfB's approach to construction industry classification had been the organization of library material. Eventually attempts were made to extend this library classification to the organization of project information (drawings, specifications, bills of quantities, etc.). A number of computerized project information applications were developed and these involved heavy modification of the library oriented classification systems; none of these application systems seem to have survived or found a permanent place in the business of building design.

While systems such as SfB and their national derivatives had much institutional support there was also much criticism; element tables are not appropriate for cost planning, materials/form tables are not useful for building products, systems as a whole are not useful for library classification, systems as a whole are not useful for project documentation, etc. In the early seventies this criticism generated a number of government sponsored initiatives sought to address some of the reported deficiencies.

Unfortunately much of the original advantages of having *one* SfB classification system vaporised in these efforts. Not only did each country develop its own flavours, soon completely new classification systems where seeing the light. Ultimately leading to the situation of today, where dozens of different classification systems fight for dominance, which will never be achieved.

On top of that, some classification efforts have recently merged with developments in Product Data Technology (PDT), an information technology that aims at semantic representation and exchange of product data. The point is that Product Data Technology in itself also can serve as a classification system, even as a 'neutral' classification system. That is precisely the reason of the current efforts to develop one mutual classification system that serves both the needs of the traditional classification system users (cost estimators and such) and the PDT people.

### 2.2.4 *Applying the Performance Concept*

For several reasons the building and construction industry has become more interested in the functional specifications of requirements. Traditionally, requirement specifications are often a mixture of functional specifications, technical specifications and prescribed solutions. But over the last decades, a shift can be seen towards pure functional specifications in terms of performance, i.e. specifications of what the result must be capable of in terms of required characteristics.

In Dutch building practice, this shift can be recognized in the new Building Regulations and the Energy Performance Standard. In Research and Development this trend goes back to the Systems Engineering methodology that dates from the fifties and sixties, and that is often applied in areas such as mechanical engineering, aircraft engineering and the defence industry. Furthermore, the development can be found in the General AEC Reference Model (GARM) [Gielingh 1988], a data model for the AEC Industry, see 5.4.2, and in more recent work on performance of buildings by Spekkink [1992], and on performance of civil works by De Ridder [1994].

Especially the trend towards Design-Construct-contracts, as discussed above, has urged the importance of functional specifications and performance specifications. As shown by De Ridder [1994], a sound performance-based specification is a prerequisite for successful Design-Construct-projects. Since the contractor is required to develop part of the design, there is no finished design description at the time of the contract. As a result, the key technical section in the agreement between principal and contractor is now formed by the specifications.

This is because there is no finished design at the time of the agreement the design is no longer the key section of the agreement between principal and contractor, but rather the specifications.

Then what is the essence of a performance-based specification? The essence is that required characteristics are expressed in terms of objective performances. In other words, the specification specifies how a product performs on specific aspects. In order to achieve this, three conditions can be put on a performance specification: a performance specification must be (1) quantifiable, (2) verifiable and (3) functional.

*1. Quantifiable* - preferably, a performance specification is defined in a quantitative way. E.g. strength performance is expressed in quantified loads that a construction must be able to resist.

*2. Verifiable* - this means that the expected characteristics of a design can be predicted, and that the actual characteristics of a realized design can be measured.

*3. Functional* - as said above, the specification of an object must specify *what* it is capable of - *not how* it must be designed or constructed. In this way, the designer has full freedom in design, and can come up with unexpected but attractive solutions.

Several authors have elaborated the performance-based approach in building and civil engineering. Below, the work of two authors that have worked on the application of this approach for building and construction, will be discussed further.

Spekkink [1992] has explored the implications of a performance-based approach for architectural design. In his work he has formulated definitions and general statements on the subject, and he has elaborated the following overview of requirements and constraints for architectural design:

1. Usage Requirements (non quantifiable)
2. Functions and Performance (quantifiable)
    - Functionality
    - Building Physical Conditions
    - Safety
3. Image/Appearance Expectations
4. Internal Constraints
5. External Constraints

The Usage Requirements (1) are the requirements as expressed by the client; they are translated into required Functions and Performance (2), that must eventually be quantified. The Image/Appearance Expectations (3) are difficult to quantify, but as Spekkink emphasizes, they must be made explicit in a design program, in order to stimulate discussion and prevent misunderstandings on this subject. The Internal and External Constraints (4) and (5) include time and money constraints, as well as regulations etc.

For an exploration of a performance-based approach, we will elaborate group (2) Functions and Performance. This group contains the following items:

| *Functionality* | *Building Physical Conditions* | *Safety* |
|---|---|---|
| Space Capacity | Temperature | Stability |
| Accessibility | Humidity | Fire Safety |
| Relational/Logistic | Air | Usage Safety |
| Extendibility/Flexibility | Light and View | Burglary Safety |
| Special Facilities | Acoustics | Disaster/Explosion |
| Durability | Vibrations | |

Based on this, an elaborate checklist is then presented for the subsequent design stages, which shows what requirements and constraints have to be dealt with in the various stages.

De Ridder [1994] has put more emphasis on the conceptual development of quantifiable requirements from an initial design goal. This works as follows: De Ridder starts with top level requirements that define the design goals, using concepts such as effectiveness and efficiency. Next, he proposes to decompose these requirements until they can be quantified. He describes such requirements using the concept of "Aspect Systems", that are defined as clusters of requirements that can be quantified, see Figure 2.1.

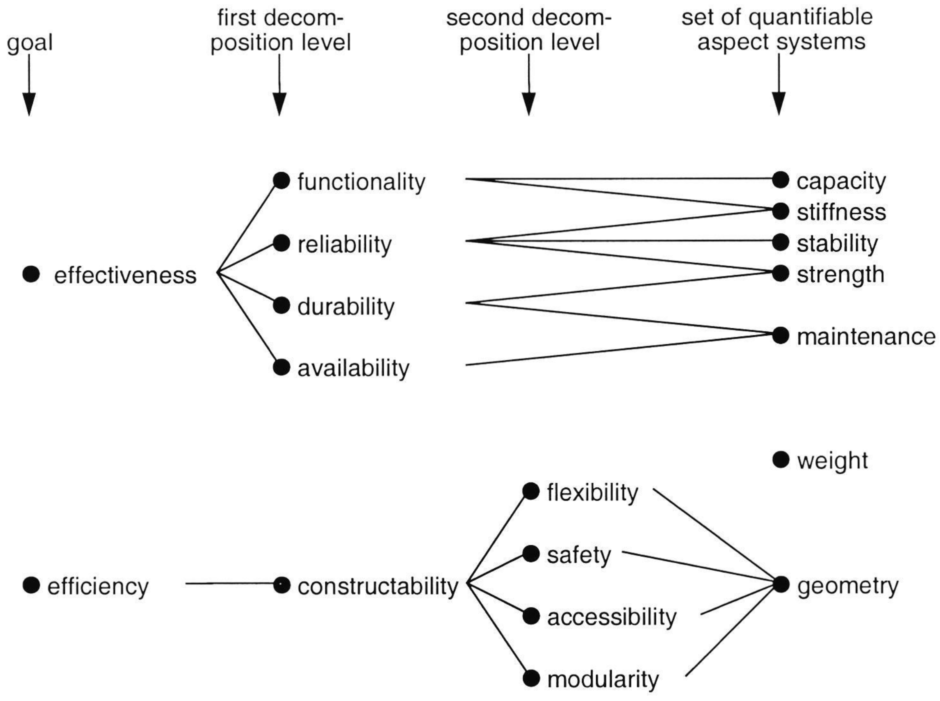


*Figure 2.1. The decomposition of the (design) goal defined as the product of effectiveness and efficiency, leading in three steps to a set of quantifiable "aspect-systems" (De Ridder). This set is meant for civil engineering, other disciplines will lead to other sets.*

## *2.2.5 Applying Systems Decomposition*

Communicating about a building or construction design usually requires large amounts of data to be exchanged. This is because a building or construction design has to be described with a large amount of data. For that reason it is necessary to structure building and construction design information. A structuring mechanism often used is *decomposition*, i.e. a structuring mechanism in which objects decompose into smaller objects.

Decomposition can be done in different ways. A widely accepted decomposition method is the so-called *systems decomposition*. In systems decomposition a building or construction is considered as an object in which multiple systems can be identified. But then again, systems decomposition can be applied in different ways. One of these is described above: the decomposition of a design goal in aspect systems as proposed by De Ridder.

In this section different existing methods for systems decomposition will be explored.

### *Systems, Requirements and Solutions*

Many definitions of the term "system" exist. In this research, we will start with a working definition by the International Council on Systems Engineering [INCOSE 1998]: "an integrated set of elements to accomplish a defined objective".

In this definition two important characteristics of systems can be recognized:

- A system is a set of elements, i.e. a set of (often physical) things;
- A system must accomplish a defined objective. In other words, a system can be related to functions, functional requirements and performance.

In other words, a system can be seen as both something "functional" and something "physical".

The next question is: how to decompose systems? According to the definition above, one could think of several decomposition methods:

- A requirements-driven decomposition, in which system requirements decompose into subsystem requirements and so on,
- A solution-driven decomposition, in which a system decomposes into e.g. assemblies, subassemblies, components and parts,
- A combination of both.

Several authors have worked on decomposition methods that support both the functional and physical nature of systems. For example, Gielingh [1988] developed the so-called Function Unit/Technical Solution-decomposition. A Technical Solution (TS) decomposes into lower level Functional Units (FUs). For each Functional Unit a set of Technical Solutions are defined, which again decompose into smaller Functional Units and so on. This approach is explored further in 5.4.2.

A slightly different approach, originating in the aircraft industry, but also applicable in other industries, is proposed by ADSE [1996]. ADSE presents two hierarchical trees: a Requirements Tree and a Solution Tree. In the Requirements Tree the project requirements are defined: at the top the highest-level requirements are found, which decompose into lower level requirements and so on. In the Solutions Tree the proposed solutions for the requirements on the different levels are defined.

This means that the design process primarily jumps between the trees as follows: from top level requirements to top-level solution (the most global description of the design), from top level solution to level 2 requirements, and so on.

This is shown in a simplified way in Figure 2.2.

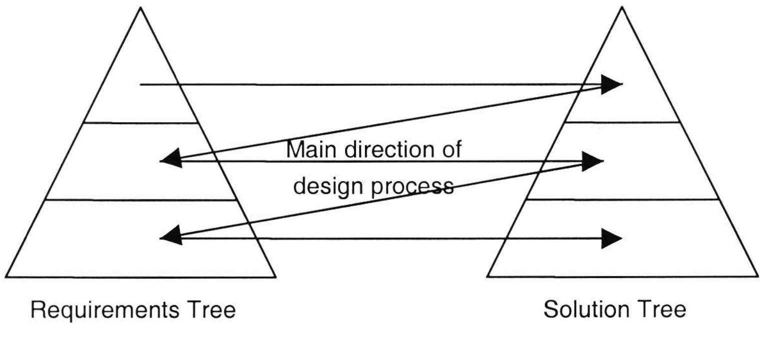


*Fig 2.2 The Requirements tree and the Solution Tree [ADSE 1996]*

The main direction of the design process as shown by the arrows, is of course not the only direction. Several feedback processes occur also, such as verification and validation processes. These processes can easily be added to the figure, but they are left out here for clarity.

Again slightly different from the trees shown above are the so-called Value Tree and Cost Tree as presented in De Ridder [1999], based on the project constraint that the needed cost should not exceed the required value of the design.

## *Different Kinds of Systems*

Systems can be decomposed in different ways, leading to different kinds of partial systems. In 't Veld [1983] distinguishes:

- Subsystem: a subset of elements while maintaining all relationships,
- Aspect system: a subset of relationships while maintaining all elements,

For any kind of system one could still ask the question how such a system should be decomposed. A general answer is: choose the system parts in a way that closely related elements remain together, while more loosely related elements may fall in different system parts (subsystems or aspect systems). But there are also formal methods for decomposition of systems. Such methods can be found in graph theory and in cluster analysis methods (see [De Ridder 1994], appendix 3 and 4, for examples and further references).

## *Elaboration and Discussion*

The next question is: how must the concepts presented above be elaborated for building and construction? The theoretical concepts described above may seem rather simple - experiences have learned that they are not so easy to elaborate in a building and construction context. See for example (1) De Waard [1992] who has been working on an FU/TS-decomposition of residential buildings, and (2) the HSL Specifications [1999], an attempt to apply the Requirements Tree and Solutions Tree.

In this section we will discuss the application of the system concepts in the context of (1) human beings and (2) buildings. This will be done in a bottom up fashion: first some examples, then an attempt to generalize.

## *Subsystems*

A trivial method to decompose a system is to cut it into pieces in a geometrical/ topological way. For example:

- a human being is decomposed into head, body, arms and legs.
- a building is decomposed into building blocks or sections, sections into storeys, and storeys into rooms and corridors.

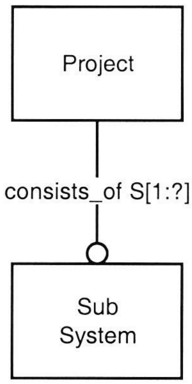


*Figure 2.3. Subsystems decomposition*

This is clearly an example of subsystems decomposition, in the terminology of In 't Veld: subsets of elements are selected, while maintaining all relationships. Furthermore, this decomposition method can be regarded as shape-driven.

## *Aspect Systems*

Another method that is often used is to decompose a system into groups of elements that play a similar *role* in the system. In other words: elements are grouped together which turn out to be closely related when a specific aspect is regarded (for example strength).

For example:

- In a human body: a muscle system, a nerve system, a skin system, a bone system (skeleton), a blood vessel system, etc.
- In a building: a vertical load bearing system, a heating installation, an electrical installation system, etc.

In this decomposition method a subset of elements is regarded, but also a subset of relationships. For example: when looking at a heating installation, we are normally not looking at its load bearing properties. Moreover, the filtering of relationships helps us in defining these kinds of systems. So in the terminology of In 't Veld, these are sub-aspectsystems. However, in this research we will refer to such systems as *aspect systems.*

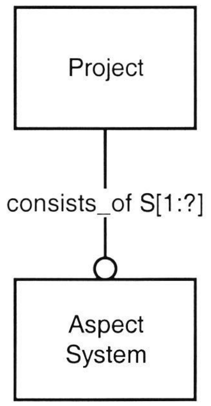


*Figure 2.4. Aspect systems decomposition*

In order to further clarify our use of aspect systems, a few common characteristics of our aspect systems are given:

- Elements of an aspect system are usually connected to each other (for example in a network structure, with nodes and links), and usually share some characteristics (e.g. material),
- While subsystems can be obtained by "cutting", aspect systems often can be obtained by "peeling",
- The role that aspect system elements share can often be described in terms of a contribution to a specific function.
- Our aspect systems are limited to the set of elements that share only *one* relation, like 'transports blood to' for the blood vessel system, or 'transports hot water to' for the heating installation system.

However, aspect systems are not considered as *functional systems* (see below), as aspect systems normally do not carry out a specific function on their own. In fact, aspect systems have more in common with subsystems as described above, since subsystems also often contribute to some function.

## *Functional Systems*

In functional systems decomposition, an object is divided into functional systems, each of which carries out a specific function. For example:

- A human being is divided into a motion system, a sensing system etc.

- A building is divided into a structural system, a thermal system etc.

A typical characteristic of a functional system is that it has a certain *performance*. Therefore a functional system contains all objects that contribute to this performance.

Note that the term *functional system* is often used in a less strict way than above. I.e. many people also use the term functional system for subsystems and aspect systems as described before. In that case, functional systems are regarded as systems that make use of other systems (as "parasites") in order to enable the functional performance.

The functional systems approach is especially popular in mechanical engineering. In building and construction a functional systems approach can be applied as well, but often leads to many overlaps between functional systems. This is because, more than in mechanical products, building and construction objects often perform multiple functions. For example, a wall in a building often plays a role in the structural system, the thermal system and in the functional space system.

For that reason, it seems more appropriate to call the functional structures that have a specific performance functional *"view models"*. In other words, the functional structures are regarded as dedicated descriptions of the building or construction, as seen from the viewpoint of a specific *discipline*.

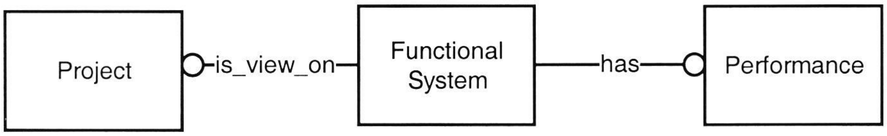


*Figure 2.5. Function systems are discipline views*

## *The Relationship between Subsystems, Aspect Systems and Functional Systems*

Aspect systems decomposition and functional systems decomposition, as presented here, show a subtle but important difference: functional systems perform a function, leading to a performance, while aspect systems at most *contribute* to one or more functions and performances.

The difference can be clarified by a comparison between a heating installation (an aspect system) and a thermal system (a functional system). A heating installation provides heat in a building. But whether the heating installation is sufficient, in other words, whether the heating installation is able to meet the thermal requirements depends on the context, for example the insulating properties of walls, windows, etc. In the thermal system description, all the relevant context elements are defined as well, thus enabling a complete performance evaluation.

The relationship between subsystems, aspect systems and functional systems, as used in this research, is shown in Figure 2.6.

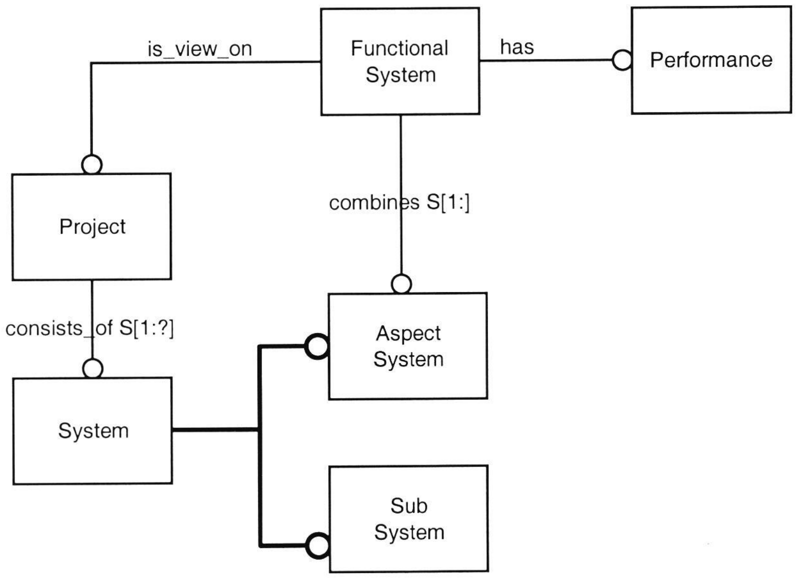


*Figure 2.6. The relationship between subsystems, aspect systems and functional systems as used in this research.*

## 2.3 Conclusions

Poor interdisciplinary communication, i.e. miscommunication, has been recognized as one of the main sources of failure and rework, in the Building and Construction industry for many decades, especially in large scale international projects. This is caused by a number of common characteristics of the industry: fragmentation, the one-of-a-kind character of both the required product and the team that realizes the product, the traditional sequential project organization and the low technological level of the industry.

A number of non ICT related developments to improve interdisciplinary communication took place in the past. Each of the approaches discussed above is a step in the right direction, but not a step that should be taken in isolation. Design-Built without a performance based systems approach is not going to do the job [de Ridder, 1994]. Common classification systems have to provide the required semantic integrity. Changing the project cultures, following the ideas of Lean Construction is required. Finally, in order to cope with the complexity in large projects, a systems approach and systems decomposition are required.

But as said above, a successful approach for improved communication must integrate the all approaches discussed in this chapter. Such an integrated approach may well benefit from developments in ICT. Therefore the state of the art of electronic communication in the Building and Construction industry will be discussed in the next chapter.

# Electronic Interdisciplinary Communication

*This chapter analyses the state of the art of electronic interdisciplinary communication in building and construction. The situation in the building and construction is compared to other industries, and factors that hamper the implementation and heavy use of electronic interdisciplinary communication in the Building and Construction industry are discussed.*

## 3.1 Introduction

In the UK, the Latham report [Latham 1994] stated that an improvement of effectiveness and efficiency in construction of up to 30 % could be achieved through process innovations. It is reasonable to assume that a significant part of this improvement could result from information and communication technology (ICT) [Wix and Liebich 1998]. Especially when ICT is used in combination with new communication approaches as described in the previous chapter.

Why, if these experts are right, is the development of ICT in the building and construction industry not coming off the ground?

Before answering this question, first the state of the art of ICT in the building and construction industry is analysed.

## 3.2 State of the Art of ICT in Building and Construction

The current state of the art of ICT in building and construction can be characterized as mainly based on Computer Aided Design (CAD)-technology. A key role is played by CAD-systems, such as AutoCAD, and data exchange often means exchange of CAD-data. Meanwhile, some integration exists with engineering (CAE) systems, building specification programs and other kinds of systems.

*However, as is often stated before, most CAD-systems are in fact not design systems, but drawing systems. At least, most CAD systems are used as drawing systems. This means that the traditional work process is in essence not changed. The drawing board is replaced by a CAD-system, but the process and the results are basically the same. Design data is represented as geometric data, the result is a drawing, which is still often plotted on paper before it is exchanged to others.*

In this situation, the meaning implied in a drawing is only stored in an implicit way. As a result, the meaning exchanged by a drawing must be interpreted by an expert. Only an expert can recognize a set of lines as a column.

From the perspective of automated data exchange, this is a poor situation. Since the meaning of the geometry oriented data is not explicitly stored, a receiving computer system will in general not be able to recognize the meaning of the data. For example, a structural engineering program can usually not automatically recognize structural elements (beams, floors) in a CAD-file.

In order to enable this kind of meaningful data exchange, researchers have started some 15 years ago to develop approaches for the meaningful representation of product data. This has led to the development of the ISO-standard 10303 "Product Data Representation and Exchange", commonly known as STEP [ISO 1993].

The goal of STEP was (and is): to provide a standard for representation and exchange of meaningful product data: not just geometry data, but also data on material, product structure and connectivity, structural and physical properties and so on. The STEP initiative was not directed towards a specific industry, but it was accompanied with a number of industry-specific efforts. For example, industry-specific standards based on STEP have been developed for the process industry, for shipbuilding and for the automotive industry, as will be discussed in 4.2.1.

For building and construction, only one STEP-based standard exists today, AP 225. This so-called Application Protocol can be used to exchange explicit geometry of

Buildings and Building Elements. A number of other initiatives have been undertaken, as will be discussed in chapter 5, but none of these have come even close to a standard.

Apart from STEP, there are of course a number of other approaches that could have helped to improve electronic interdisciplinary communication in building and construction. In chapter 4 some approaches will be discussed that are based on newer and better technologies such as CORBA and Internet. But also these new approaches have not led to improvements in building and construction as yet. And it is very unlikely that this is going to change soon.

So let us go back to the question at the beginning: Why is the development of ICT in the Building and Construction industry not coming off the ground?

A number of reasons spring to mind. Some have to do with the organization of the Building and Construction industry, some follow from the organization of the ICT industry, and some are more of an ICT technical nature. The next section describes the problem in more detail.

## 3.3 Why Electronic Interdisciplinary Communication Is Lacking

### *3.3.1 B-C Organizational Reasons*

The most obvious reason is that the Building and Construction industry is too fragmented. So many players, so many national differences, so many interests.

A second reason is the one-of-a-kind character of the Building and Construction industry. Every construction work is unique, at least because every site is unique. But also project teams are normally set up differently in every new project.

Another reason is that there are no large and influential international players, as in other industries as Automotive (see below). The best-organized parties in Building-Construction, the national governments, are not playing on the international stages. This is surprising when one thinks of the potential 30 % improvement by process innovation as claimed in the Latham report.

Why are national governments in Civil Engineering and Public Works not jumping on this bandwagon? Why are large facility owners not keen on pulling resources together to make things happen?

The reason is that national governments are not interested in international projects because they have to take care of their own backyard. Money spent on international standardization cannot be justified in terms of return on investment and might even profit other countries more than their own. And ‘others’ should take care of these problems, such as the European Commission in our case. The European Commission is (or perhaps better *was*) indeed more favourable towards PDT-standardization projects. Not that funding is directly given to STEP standardization efforts, but projects that do have budgets for standards development (PDT standards) have better chances of getting funded.

Yet another factor in this context is that most initiatives have come from design-oriented groups. Unfortunately, designers are normally not very powerful parties in building and construction, and usually not strong enough to force other parties to invest in specific innovations.

Some countries have seen the ‘light’ and are investing in PDT, not in ISO-STEP, but in national projects. The CORENET project in Singapore is an example, but by no means the only example. Most governments try to protect their own national B-C industries. Unfortunately all this protectionism is in vain, national efforts, without international back-up are quite useless and a waste of money.

### *3.3.2 ICT Organizational Reasons*

ICT is an international marketplace. Vendors of the Computer Aided support systems are mainly based in the USA, so purely national efforts to solve the integration problems are of limited value outside the USA. But the same is true for ICT vendors from other countries, like Germany, or the UK. Only selling in one country or one region is not an option for a healthy company.

Another aspect is that ICT vendors that compete in the same market are not always willing to adopt common standards, as clearly demonstrated by Microsoft. Lucky enough, however the situation is not as bad as could be, organizations like OMG (Object Management Group) and W3C (World Wide Web Consortium) are industry initiatives that do provide us with a lot of good standards for the Internet and such.

### *3.3.3 ICT Technical Reasons*

Several, largely incompatible, efforts in developing standards for *inter* discipline communication are being developed, sometimes based on quite different technologies.

ISO-STEP, the badly financed main player in the standardization field, is (1) using already somewhat outdated information technology and (2) is not doing too much for the Building and Construction industry anyway.

## 3.4 Lessons from Other Industries

### *3.4.1 Shipbuilding*

Though ships are comparable to Building and Civil Engineering structures, the Shipbuilding industry is quite different. Not only is it better organized, with only a limited amount of players, it also has the big advantage of well established bonds between designers, engineers and production people, and, last but not least, production on a yard is quite different from production on a site.

These differences reflect the way Shipbuilding takes up ICT. Starting with the European MARITIME-project the Shipbuilding industry developed its own STEP extension, called Building Blocks (see next chapter for details).

### *3.4.2 Process Plants*

Also the Process Plant industry is quite different from Building Construction. Its large and rich international Clients, like Shell, demand and enforce standards for electronic communication, including interdisciplinary communication. The solution of the Process industry is also based on a STEP extension, but different from the Shipbuilding industry (AP221, Epistle).

### *3.4.3 Automotive*

The Automotive industry again is quite closely organized and has been able to develop electronic communication standards with ISO-STEP (AP 214). And again the solution differs from both the Shipbuilding and Process industry.

## 3.5 Conclusions

Compared with (some) other industries the Building and Construction industry is extremely fragmented. Furthermore, its most powerful players, national and regional governments, are not performing on the international stage. Other industries, with comparable product complications, have partly solved the problem. In most cases a solution was found within the STEP development, but amazingly enough, each solution is different.

Which conclusions the Building and Construction industry should draw from the above is not really clear. First of course the well known fact that, on an international level B-C is not very powerful, and second, that the ISO-STEP standardization process seems increasingly inadequate. Threatening is the fact that related industries like the Power Plant and Process industries (which include B-C works) are ahead of us, and after some time will demand B-C companies to communicate in their languages.

Maybe the right conclusion is that interdisciplinary communication in B-C will not come about in the same way as in the other sectors. Maybe the idea that we first have to develop and accept international standards for project communication that subsequently have to be implemented in a large number of ICT vendor systems, that subsequently have to be sold and used by our industry before BC can start to change its project Information System does not suit the nature of our industry.

# Available Integration Technologies

*This chapter examines the various basic integration technologies currently available for the development of interdisciplinary communication for large-scale building and construction projects.*

## 4.1 Introduction

There are several technologies available that can be used to implement interdisciplinary communication. Some are more open than others, some are more meaningful than others, some are fast and some are slow, some are coming and some are already on the way out.

Without going in too much technical detail, the chapter gives an overview of each of the technologies (and their support) and analyses its applicability and limitations for implementing interdisciplinary integration for large scale building and construction projects.

Generally spoken, the most important developments seem to be:

- The still ongoing work on the STEP-standard,
- More recent developments based on object oriented technology, especially the OMG-CORBA-related work,

- The fast moving developments based on Internet technology.

However, these developments are by no means taking place separately. In fact the most interesting developments are the ones in which different technologies come together. For that reason, this chapter will discuss developments more or less in the sequence given above, but will zoom into "multi-technology" developments from time to time.

## 4.2 Overview of Technologies

### *4.2.1 ISO 10303 STEP*

The International Standards Organization (ISO) is a non-profit organization that organizes the development and maintenance of international standards. ISO 10303 [ISO 1993], commonly named STEP, is the standard that supports the industry-wide exchange of product model data between different vendor platforms and applications. The development of STEP started already in 1983 and can be seen as the leading international effort in standardization of product model data. Currently most high and medium tech industries have one or more STEP standards, called Application Protocols (APs), in use.

#### *The STEP Integration Methodology*

The current STEP integration methodology started from the idea that user defined Application Reference Models (ARM) can be transformed into more common Application Interpreted Models (AIM), by incorporating common constructs found in STEP Resource Models. There are different types of Resource Models, covering most of the entities that are common for each application world. This process of redeveloping an ARM into an AIM - sometimes called the interpretation bottleneck - is quite cumbersome. First it has to be done by a small team of USA and UK based individuals. Second it is expensive[5]. Third it obscures the original view on the data in the ARM so that the community it has to service often no longer recognizes it. And finally, integration is only locally defined within the context of the AIM version of the model. Information exchange between different Application Protocols is not possible.

[5] Maybe one and two are related!

The restrictions discussed above made the current STEP integration insufficient for most industries with a large amount of shared data. Some industries found the resources to develop an extension of the STEP methodology that worked within limits. Some of these efforts will be explained in the next sections. The Building and Construction industry made one small and badly funded effort with the development of the Building Construction Core Model, which also will be discussed below.

### *Evaluation*

Though the original STEP methodology has been augmented by several constructs, such as Application Interpreted Constructs (AICs), it is still not really suitable for the Building and Construction industry.

### *Building Blocks*

The Shipbuilding industry was among the first to recognize the STEP integration problem and developed the Building Block approach as a solution [MARITIME 1995]. Building Blocks (BB) are schemata that partition one large information model for ship design into some 40 smaller parts. Each BB describes the notions of interest of one particular subject, e.g. piping. Each BB 'knows' its interfaces with BBs that are somehow related. All the BBs are embedded in an Object Oriented 'sauce'.

### Evaluation

The BB approach is successful in the Shipbuilding industry because of the restricted number of players in that field. The object oriented embedding is nice, because it enables co-operation between separate BBs, however at the same time BBs are not integrated in the mainstream STEP AIMs. This, like all efforts of its kind, makes the BB environment into an island of integration.

### *AP221/Epistle Model Family*

Also the Process Plant industry developed its own STEP extension. The approach followed here is to use a common meta model as an extension of the STEP methodology. The meta model defines a structure for a large variety of information groups.

Figure 4.1 below shows a small part of the Epistle/AP221 meta-model [Teigeler, 1996]. The model, inspired by the CYCL tool, shows that the world consists of a collection of 'things', that can be categorized in three dimensions: (1) fact/idea, (2) type/instance and (3) what it describes, i.e. a physical object, or something else. Each specialized 'thing' can be specialized further.

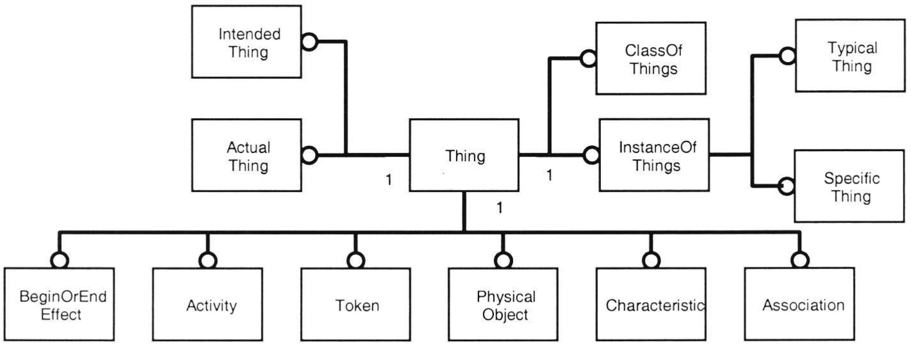


*Figure 4.1 The Epistle/AP221 metamodel. A classification of things.*

## *Evaluation*

The PP industry is rapidly moving forward. Integration based on the AP221/Epistle model family is being implemented in practice. Recently the POSC/Caesar (oil exploration and production) joined this effort. The success mainly comes from a dedicated small group of rich clients. As with BBs, the PP approach also creates an island of integration, because only exchange within the AP221/Epistle scope is supported. It should be noted that the group is expanding. The offshore industry recently joined the group. If the Building and Construction industry will follow is not sure.

Although the Dutch CROW has taken the initiative to apply the AP221-approach for civil engineering, it is doubtful whether the approach will work in Building and Construction. The problem is, again, the fragmentation of the industry and the lack of powerful parties that really push such a development.

And then things become even worse: in the future Building and Construction will have to cope with integration problems created by the AP221 island, because buildings and civil engineering works are also part of Process Plants.

### *AP214 Mega AP*

The Automotive industry also found a solution to the STEP integration problem. The solution adopted is the development of a 'mega' AP, AP214, with thousands of entities.

### *Evaluation*

This approach is also successful because a small team of rich car manufacturers did the job. Again not an example BC can follow.

### *BCCM*

One effort that took place in BC was the development of STEP Part 106, the Building Construction Core Model [ISO 1994]. This development, performed by Tolman and Wix, followed a decision taken by the STEP AEC group to adopt the idea of layered modelling using core models. The decision was inspired by the successful demonstrations of the European ATLAS-project (see chapter 5).

### *Evaluation*

The BCCM work continued for 2 years before it lost its (financial) support. After that the International Alliance for Interoperability (IAI) started the development of the Industry Foundation Classes (IFC) on a (somewhat) more commercial basis. The IAI-IFC took over the BCCM development. See chapter 5. This effort clearly shows that ISO STEP is not the right platform for BC. Or, alternatively, that ISO-STEP is not so much about standardization, but about pre-standardization research.

### *Components*

As discussed above, the STEP development is struggling with its own complexity. The development of a STEP Application Protocol turns out to be a long-lasting, expensive and cumbersome job. And integration of Application Protocols, or AP interoperability in STEP terminology, is an old but still unsolved major issue.

In an attempt to overcome the difficulties in STEP AP development and interoperability, several groups have searched for a solution based on some kind of modular approach. The general idea is that it must be possible to define small model parts, develop such parts independently and enable the integration of the parts by proper interface definitions.

The component approach, often called Business Objects, is used in most Enterprise Resource Planning (ERP) systems. The European WONDA project [World wide enterprise data interoperability, EP 25.741, http://www.bild.ie/wonda] researched the concept in a Building and Construction industry setting. The main advantage is that it offers a cheap solution for the integration problem that is incrementally extendable. The concept also supports different views. Within the STEP organization, Working Group ISO TC184/SC4/WG10 is developing a components-based approach that could be added to the official STEP methodology [Staub & Grabowski 1999].

### *Evaluation*

The components-based approach as being developed in ISO TC184/SC4/ WG10 promises to solve some of the severe problems in AP-development and interoperability. As such, the approach could result in a major improvement of the STEP methodology. Moreover, it could even be an improvement that is necessary for STEP to stay alive, if survival is still possible.

Component technology in general will be important in the future. Integration with ERP systems is required. Various enabling technologies like ActiveX, JavaBeans and others are already around.

### *STEP Future*

At the moment STEP is in a process of change. Too many restrictions and tinkering severely slowed down the interest in STEP developments for the Building and Construction industry. A new STEP Architecture with a more modular approach is being developed.

Whether STEP will play a major role for BC in the future is doubtful. Basically it is not so much the technology that is the limitation, but the lack of commitment of the industry itself.

### *4.2.2 ISO 13584 PLib*

ISO 13584 PLib [1995] is another ISO-standard for industrial applications. PLib focuses on the development of electronic parts libraries. The scope of ISO 13584 also includes the provision of mechanisms and definitions necessary to enable property information on parts; such as information about names of the manufacturers, uses, conditions for use, geometry, materials, performances, behaviour, and safety. PLib parts information can be exchanged between different computer systems, so that it can be utilised for search and selection of parts.

Important for this study is that PLib also supports the notion of different (shape) views on a part. Each part can be represented with different shape models, including models for FEM analysis.

PLib is also related to STEP. It is possible to use PLib objects in STEP product models (see Part 101 of ISO 13584).

#### *Evaluation*

PLib fills a niche that the STEP standard does not deal with: the modelling of standard parts libraries. As such, PLib offers new possibilities for modelling approaches based on standard parts.

But, as PLib is rather complicated and more suited to libraries of Mechanical parts then construction products, it seems doubtful that PLib will play a major role in BC.

### *4.2.3 UN-EDIFACT*

Edifact is a standard for encoding EDI (Electronic Data Interchange) messages issued by the UN. EDI concentrates on business data, i.e. procurement.

#### *Evaluation*

Integration of business data and project data will be important in the future. Several attempts to integrate both worlds have been made. Probably XML, the next Internet language, will be used successfully for this purpose.

### 4.2.4 OMG – CORBA

The Object Management Group (OMG) developed a standard and mechanism for object sharing and exchange, called CORBA (Common Object Request Broker Architecture). Objects are defined in the Object Oriented sense as: the encapsulation of the attributes, relationships and methods of software identifiable program components.

The OMG consortium was founded in 1989 as an independent non-profit corporation. The consortium now includes over 800 members, including all the important ICT-vendors.

Using OMG CORBA allows dynamic aspects to be taken into account. This not only means that windows and doors that can be opened and closed, but mainly that we are no longer restricted to sharing *data* models, but that we now can share *object* models. Object models include behaviour and can be used for real time control. If for example the architect changes the height of a building the consequences for the different participants are immediately shown in their view on the model.

In recent years, many research efforts have been using CORBA and OMG technology in order to make steps forward in specific application areas. A more or less building and construction oriented example is the EC-project VEGA [VEGA 1998]. In this project an attempt is made to link the data-oriented STEP technology and the object-oriented CORBA/OMG, in order to integrate product data technology with workflow management and OO technology.

#### *Evaluation*

One of the advantages of CORBA for BC is the possibility to share a dynamic, distributed model of the artefact (and its construction process) for instance for co-operative or concurrent design/engineering. For BC with its numerous participants object sharing will be very important in the future.

### 4.2.5 Product Data Management (PDM)

Product Data Management (PDM) is a technology that focuses on the organization and management of product data, e.g. data held by CAD drawings and technical reports.

Compared to Product Data Technology (PDT), PDM puts the emphasis on the organization and maintenance of the representations of product data, rather than on the contents.

This means that PDM is dealing with things like:

- Organization of CAD drawings (and calculations, technical reports etc),
- Definition of workflows around CAD drawings,
- And last but not least, links with Enterprise Resource Planning (ERP) systems such as Baan and SAP.

On the other hand, PDM hardly deals with the data that is represented in the CAD drawings and reports.

To a certain extent, PDM can be seen as a mixture of Product Data Technology and Electronic Document Management (EDM).

From an information modelling perspective, PDM systems have a relatively simple structure. Normally, the product data (the drawings and the documents) are structured around simple product trees, decomposition trees of products and parts. These product trees serve as directory structures, in which the documents are stored. On top of this structure, workflow mechanisms, change control procedures etc. can be built.

Obviously this data structure offers only limited possibilities to implement product modelling concepts as aimed in STEP and Product Data Technology. But some vendors of more advanced PDM-systems are participating in STEP and are working on this issue, for example Sherpa Corporation and EPM Technology.

Unlike most of the other technologies discussed in this chapter, PDM is a technology that is mainly developed by commercial software vendors. This has resulted in a number of commercially available PDM-systems. Examples are Metaphase II, Optegra, Baan-PDM, MatrixOne, SmarTeam, etc.

These systems generally work well (although they are not very cheap). Problems arise in environments where standardization on a single PDM-system is not possible or not wanted. Especially in the fragmented Building and Construction industry, this is a serious problem.

The problem of heterogeneous PDM environments has been explored in the SAVE project [Bodington and Sims, 1999] and by Doblies and Rothenburg [1999]. In both projects, the problem is being tackled by using the so-called PDM-enablers by OMG. It is expected that PDM integration will mature in the coming years.

*Evaluation*

Compared to the other technologies discussed in this chapter, Product Data Management has the following advantages:

- PDM systems are commercially available and do their job properly,
- The data structure is relatively simple, allowing a straightforward implementation of simple product models.

Weaknesses of PDM are:

- The lack of openness of the commercial PDM systems, resulting in exchange problems between PDM systems,
- Limited possibilities for the definition of product model data as aimed in STEP,
- Limited flexibility and extendibility of the product data structure.

### *4.2.6 Agent Technology*

An advanced knowledge-based technology that supports improved flexibility and extendibility of PDM-systems (amongst others) is *agent technology.* Agent technology is based on the notion of (software) agents: pieces of software that can act upon the structure of (in this case) a PDM-system in an autonomous way. Preferably, agents are able to modify or extend the PDM-system in order to better support users in their product data management tasks. For example, an agent-based PDM-system might be able to extend itself with additional views on the product database.

Research on the development of software agents for PDM systems is carried out for example by Anderl and Arlt of Darmstadt University [1999].

*Evaluation*

Agent technology is still in its (intelligent) childhood. It is one of the last buzzwords in Knowledge Technology. If it will be applicable for BC and if it can play a role in solving the interdisciplinary communication problem is a question that is left for the future.

## *4.2.7 Internet Technology*

### *The Current Internet (HTML)*

The World Wide Web Consortium (W3C) has been extremely successful in its efforts to push the current HTML (Hyper Text Markup Language) based Internet. For building and construction, the role of the current Internet is limited to the sharing and exchange of documents.

### *Evaluation*

The main problem with the current Internet is that it is unsafe, slow and unstructured. HTML is much too simple.

### *The New Internet (XML)*

The new Internet will be based on XML (eXtensible Markup Language). XML will be the new 'hype'. The reason is that XML and related standards will overcome the limitations of the current Internet.

Basically, XML is a simplified form of the complex SGML (Standard Generalized Markup Language, an ISO standard). The need for something simpler than SGML was already shown by the popularity of HTML. But HTML is an application of SGML, with a predefined document type description. Therefore HTML does not support complex document structures, and does not distinguish between presentation and content.

Since XML is not an application but a simplified form of SGML, XML does support things like complex document structures, and the distinction between presentation and content. This makes XML better structured, flexible and very attractive for the communication of product data over the Internet.

One of the most interesting possibilities is to develop so-called XML vocabularies, i.e. Internet languages that are tailored to the needs of special groups of users. Two examples of XML vocabularies are OFX and CML. OFX (Open Financial eXchange) is an Internet language for secure banking and CML (Chemical Markup Language) is a language for the chemical industry used to communicate about chemicals.

XML is not a standard on its own, but embedded in a number of related developments. Examples are: XLink and XPointer which can be used for multiple and bidirectional chaining, X3D a VRML frontend, XMI a metamodel that allows us to send a schema together with its data, etc.

Currently several XML initiatives for the Building and Construction industry are underway. Bentley Systems in the USA announced the aecXML initiative. aecXML will be an XML vocabulary for the A/E/C industry that will be developed as an extension of cXML (Commerce XML). The focus of aecXML is not so much on Design/Engineering but on E-Commerce type of applications for the Building and Construction industry. Another initiative is the European E-Construct project (IST 10303). E-Construct will develop, implement, apply and disseminate an XML vocabulary (bcXML) that supports meaningful electronic communication over the national borders about components, systems, services and resources. It takes account of differences in languages and classification and coding systems.

### *Evaluation*

XML and related technologies will make Internet very popular in BC. The OFX example shows that the new Internet can be safe. The CML example shows that it is possible to raise the level of semantics, which contributes to increased speed and structure.

The idea to develop specialized XML vocabularies that can support electronic business and communication between people and computers is exciting. XML vocabularies are simple and easy. They will be quickly used for E-Commerce type of applications and allow companies and individuals to apply available PCs. There are no huge standardization efforts required and procurement of components, systems, services and resources is a major issue with all kinds of financial consequences.

The fact that this technology will not be dedicated primarily to the support of Designers and Engineers, but really supports vendor independent open communication, is probably one of the main cornerstones for success.

### *Java*

Java is a platform independent Object Oriented (OO) programming language embedded in an environment that includes components (Java Beans).

### *Evaluation*

Java technology is one of the prime candidates for the support of better electronic communication, because it is platform independent and truly OO. A problem with Java might be its execution speed, which is somewhat slow because it is interpreted.

Also very promising is the combination of Java and XML, as XML supports the communication of database schema and OO hierarchies.

### *Microsoft*

Another important factor is Microsoft with its Windows operating systems and its ActiveX technology. Strictly spoken, this is not a Internet development, but it can be seen as a commercial competitor of developments such as Java.

ActiveX is a standard for component embedding, which provides cross platform and application developments. ActiveX is more or less similar to Java Beans, but restricted to the Windows environment. There is software available that translates Java Beans into ActiveX components and vice versa.

Microsoft is also active in the XML arena, for example in the BizTalk initiative.

### *Evaluation*

The Microsoft environment is probably the most important candidate platform for the realization of interdisciplinary communication in BC; it is cheap, powerful and widely used. Moreover Microsoft is very powerful and likes to dominate as shown (once again) by its Internet activities.

Because most companies and countries fear the power of Microsoft, alternatives to BizTalk pop up. The latest player in the field is the UN. Together with OASIS the UN pushes ebXML as the open international counterpart of BizTalk.

## 4.3 Conclusions

Basic integration technologies are still evolving with great speed. What today is state of the art is already old-fashioned in 4 or 5 years. In this changing environment BC has to move forward to a more integrated way of working.

Communication technologies like CORBA and Internet have a partly additional and partly overlapping functionality. CORBA is widely supported and may well support high volume information exchange required for Client-Server architectures where 3D-CAD and FEM play an important role. Also Internet is starting to play an important role especially when HTML will be succeeded by XML.

On the standardization front things are moving slowly. Don't count too much on STEP for BC [Tolman 1999]. STEP started way back in 1983, but the practical results for building and construction are still disappointing. Moreover, the current STEP developments mainly result in islands of integration. It also seems reasonable to forget about a top down approach for our industry; there are not enough powerful players and those parties that are powerful (national governments) are not really joining the game[6]. The best bet for the near future is the IAI-IFC development, but don't overestimate its momentum, IFC members only contribute a few percentages of their marketing budgets to this development. Consequently only a small part of a building design project can be supported at this moment.

Individual countries and individual companies do not play any significant role. Investments required to keep up with the rapid changing technology are extremely high. Moreover, most ICT-vendors come from abroad and cannot be expected to conform to every national standard that comes up. Company standardization is probably worst. Once some standards are in use it is hard to replace them (and their inventors) for something better.

The final conclusions from this chapter are as follows.

- The traditional 4-step strategy to provide interdisciplinary integration is not really feasible in our industry. The four step strategy – (1) develop standards, (2) implement standards in CAxx systems, and (3) sell these systems to the BC industry and (4) start to use these systems in building and construction projects –

[6] If the Latham report is right and up to 30 % can be saved by process innovations in construction, then no government can sit still and relax.

seems to work in other sectors of the industry, but not in Building-Construction. Another strategy is clearly required.

- Basic communication technologies to develop, implement and use dynamic distributed models are becoming available. The most promising technology that really seems to suit the needs of our industry is the new XML based Internet. The combination of Java and XML (plus related technologies) will be the first candidate to evaluate for implementation purposes in the coming years.

However, at the time writing, promising Java and XML solutions were still under development. We should also keep in mind that Java and XML are languages, not ready-to-use solutions. Interdisciplinary communication is about meaning, not mark-up. In other words, it is first necessary to develop Java- and XML-based programming tools that suit the needs of the Building and Construction industry before these technologies are ready for applications. Such tools are currently becoming available.

For these reasons, technologies based on Java and XML have not been used in the implementation work done in this research. Instead, implementation work has been carried out using available STEP-PDT and PDM-tools. This does not mean that XML (vocabularies) and Java are not seen as important. On the contrary, implementation of the ideas presented in chapter 7 will in the near future benefit from these low cost, open and widely available communication technologies.

# Research on Interdisciplinary Communication in Building and Construction

*This chapter analyses the state of the art of Product Data Technology in the Building and Construction industry, mainly focusing on support for interdisciplinary communication. After a brief description of the problem the different paradigms used in the various developments will be analysed. The most important efforts will be discussed in some detail. The chapter ends with a conclusion about the current state of the art of PDT in the Building and Construction industry.*

## 5.1 Introduction

This chapter discusses the various efforts of the past and present to provide electronic interdisciplinary communication for the Building and Construction industry. The focus is on attempts that apply Product Data Technology (PDT).

PDT can be applied in a number of ways, as shown in practice. The chapter analyses the pros and cons of the different approaches and presents the conclusions of the analysis.

## 5.2 The Problem

Interdisciplinary communication in the Building-Construction industry is very important. Not only because there are a large number of disciplines involved, but also because (1) in most cases participants in a project are working together for the first time, and (2) for practical purposes each discipline uses a (mental) model of the project that is particular suited to support its own role. This means that different abstractions of the same objects co-exist in the minds of project participants from different disciplines. From an information point of view, this results in a rather complex situation. Finally in most cases interdisciplinary communication is not restricted to BC, but also involves other sectors like Mechanical, Electrical, Power and Process Plants and others.

With the introduction of Information and Communication Technology (ICT) in the Building and Construction industry each discipline and each company (or even each individual) obtains the system that is most suitable for its needs and taste. Consequently there are now numerous CAxx systems that can play a role in a project and in a company. With this proliferation of systems, the question how to provide electronic interdisciplinary communication, becomes all the more interesting and all the more difficult to answer. After a more detailed discussion of the problem, an analysis of the various solutions tried over the years follows.

## 5.3 Problem Analysis

Communication between human beings requires common understanding and agreement of syntax, semantics and pragmatics. Syntax covers the language rules. Semantics the words. And pragmatics the way the words and sentences are uttered. Electronic communication is not really different. Also the syntax and semantics cover the way meaning is captured and understood. The meaning of pragmatics is now changed to: the way meaning is translated into bits and bytes and electrical pulses, i.e. which basic integration technology (chapter 4) is used.

### *Syntax*

Computers are rather stupid; they mostly do not understand what they are communicating about. Unlike humans, their ability to abstract meaning from data is hardly non-existing. Computers with common sense are not to be expected in another decade. This means that the syntax of the communication language cannot be the same as ours. Natural language is simply too difficult for computers. One of the famous sentences to illustrate the difficulties is: "Time flies like an arrow". This sentence can mean many things, one of which is that a certain kind of flies, called "time-flies" are fond of an arrow. From this example it is already clear that communication between computers cannot yet be based on natural language syntax. A more restricted type of syntax is required.

Most information modelling language use a syntax that supports binary sentences only. An example of a binary sentence is: a "Wall" contains one or more "Windows". *Nouns* are "Wall" and "Window" and *verb* is "contains". No adjectives, adverbs etc. are allowed. Additional is the possibility to express cardinalities like 0, 1 or many. In database oriented languages nouns are used to denote entities (things of interest to be stored in a database) or sets, and verbs are seen as relations between entities.

Recently the syntax of Object Oriented design became popular. In OO Nouns are used to denote *Classes*. A Class is a group of objects with a common behaviour. OO adds a language element called *Methods* to the syntax. A Method is something that all the elements of a Class have in common. For instance all the members of the Class "2D geometrical surface" (i.e. square, trapezium, triangle) can have the Methods "Rotate", "Translate" and "Scale".

Standardising on a given syntax is not too difficult, although also here the rapid change of the technology is at least cumbersome. Shall we use a traditional relational database and a simple entity-relationship modelling language, or shall we go for an OO approach and/or a dynamic model?

### *Semantics*

When agreed on the syntax to be used in the communication, words (semantics, meaning) become important. Agreeing on semantics is very difficult and not a simple matter of choice. The following problems have to be taken into account.

Different disciplines often use different words for the same object, or the same word for different objects. Also each discipline uses a lot of words that are very relevant inside the discipline, but not, or much less outside the discipline. Also different countries use words with different meaning for the same object and the same words for different objects, not only countries with different languages, but also countries with the same language (i.e. Schedule means something different in the USA and the UK)

Another problem is that the number of words and the meaning of the words are not static, but changing. Often words are also related to the artefact or facility at hand (i.e. the meaning of a word is depending on the context).

Finally also the number of words is quite a problem. How many and which words should be taken into account? A few thousand some experts say. No more than 200 others reply. Though currently the small model advocates seem to have won the debate [Junge and Liebich, 1998], in reality the debate is still open. The reason is that small models are nice, but probably not good enough to do the job. It is not for nothing that historically BC uses thousands of words to communicate about all the subtleties involved in large-scale construction.

This big model vs. small model discussion will be continued in 5.6.

### *Multiple Functions*

Another problem is caused by the fact that objects can play a role in several systems. A classical example is a radiator (heating element) that also serves as a balustrade.

### *Interdisciplinary Communication*

Each discipline has a vocabulary that is well understood in his discipline. Interdisciplinary communication means mutual understanding. This means that a participant sometimes has to describe his own discipline-specific concepts into common concepts that are understood by the other participant. This results in a situation in which many translations and conversions take place. Of course in real life this is often a mental process; in an information system this results in a number of complex conversion mechanisms.

In AI Research, the term “ontology” is often used to describe the concepts that someone uses to describe (or model) his Universe of Discourse. Such ontology describes objects, properties, relationships, and dynamic characteristics that a human being reasons with. In this sense, participants in a building and construction project have *partly overlapping ontologies*.

Then how to deal with overlapping ontologies when Product Data Technology is applied? That is the main question that is discussed in this chapter. Over the years a number of approaches have been proposed and tried out. First of all, the STEP architecture provides the application protocol approach (see also chapter 4).

Furthermore, a number of alternative approaches have been proposed. In the next sections these will be discussed as follows. First, in 5.4, the theoretic ideas or paradigms that lie behind the approaches will be discussed. Next, in 5.5 and 5.6, the results of a few projects will be discussed in which the paradigms have been used. Finally, some conclusions will be drawn.

## 5.4 Different Paradigms

In this section a number of approaches or paradigms towards overlapping ontologies in multi-disciplinary engineering design are analysed.

A starting point in this analysis is the objective that communication in building and construction should be based on concepts that are commonly used by building practitioners, such as walls, columns, etc.

Similarities and differences between existing approaches have been subject to a number of earlier analyses, for example in [Björk 1995], [Hannus 1995] and [Tarandi 1998]. For that reason, this analysis will not explore all existing approaches in great depth. Rather, the analysis will concentrate on the key characteristics of a few important (influential or challenging) approaches.

### *5.4.1 The STEP Application Protocol Paradigm*

For information needs of different engineering disciplines, the STEP standard (see 4.2.1) uses so-called Application Protocols (APs), in which application-specific data are defined that must be exchanged. As a result, exchange between different disciplines means that information must be exchanged between APs. In other words, interdisciplinary communication must be accomplished by AP-interoperability.

Unfortunately, AP-interoperability is a major STEP issue for a long time. A good analysis of the problem can be found in [Staub & Grabowski 1999]. In short, the problem is that:

- some standardization of APs exist because of the Generic Resources,
- some ad-hoc interoperability is achieved by so-called Application Interpreted Constructs (AICs),
- other approaches have been developed, such as the Building Block approach as proposed by the shipbuilding committee, but these are not part of the official STEP methodology.

- a number of practical problems occur in efforts to integrate APs, such as the complexity of individual APs, the so-called integration bottleneck (see 4.2.1), long-lasting and a-synchronous development schedules of APs, etc.

But most importantly, the STEP methodology leads to information exchange between APs on the abstract and rather meaningless level of "product items", "geometry items" etc. This is because AP-concepts must be "interpreted" as Generic Resource concepts (such as "product item"). Exchange of *meaningful building concepts* such as walls and columns is not supported sufficiently.

## *5.4.2 The Functional Unit/Technical Solution Paradigm*

The next paradigm that is discussed here is the Functional Unit/Technical Solution-paradigm as proposed in the General AEC Reference Model (GARM) [Gielingh 1988].

In the GARM, a distinction is made between functional and physical concepts, and these concepts are referred to as Functional Units (FU) and Technical Solutions (TS).

- Functional Units describe objects "as required" and have Required Characteristics.
- Technical solutions describe objects "as designed" and have Expected Characteristics.

This approach makes it possible to *evaluate* a design by comparing the expected characteristics of Technical Solutions with the required characteristics of the associated Functional Units.

The GARM also proposes a decomposition tree in which:

- TSes decompose into lower level FUs,
- these FUs may have TSes,
- these TSes decompose again into lower level FUs

etc., see Fig. 5.1.

The GARM not only defines the concepts of FU, TS and decomposition, but also the concept of Interface. Functional interfaces are specified between the FUs on one level of decomposition. These functional interface requirements have to be met by the actual technical interface solution provided by the collective set of chosen TSes.

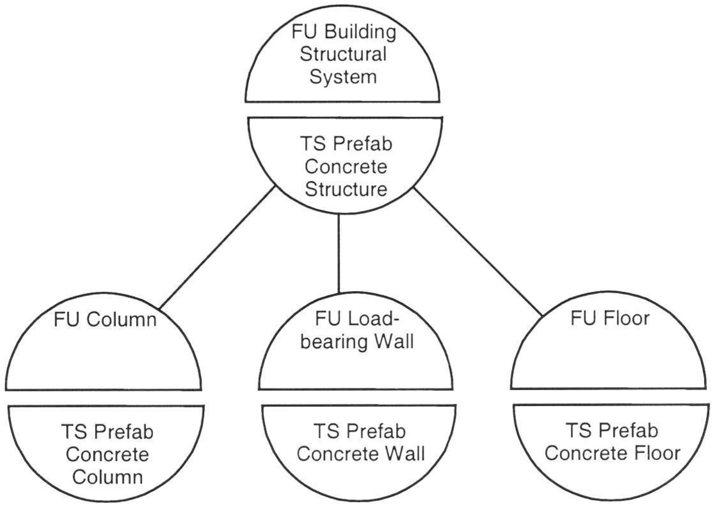


*Fig 5.1: FU/TS-decomposition (GARM)*

One of the objectives of this decomposition approach is to provide a structure for design management. For example, a lower level Functional Unit (and its functional interfaces) can be given to a small team of specialists that is asked to find a Technical Solution for the requirements that are stated in the Functional Unit. By doing so, the Functional Unit becomes the concept that is exchanged between design partners.

## *Evaluation*

The GARM and the Functional Unit/Technical Solution paradigm have had a major influence on building product modelling. It has also been subject of many discussions. An often-stated criticism, especially by architectural design researchers, is that the GARM in fact prescribes a functional design method ("design as problem solving", comparable to design approaches such as presented by Alexander [1964]), which is rather rigid and might limit design creativity. But in most design processes, the dealing with function and functional requirements is a key aspect of design, as already discussed in 2.2.4. And still there are numerous examples in practice of design projects that are supposed to be rational and function-driven, where functional requirements are not made explicit, and where the checking and validation of the design still is a matter of instinct (sometimes called "professional judgement").

All in all, the FU-TS paradigm is quite powerful and applicable to our industry. It supports both top down and bottom up design and is also applicable in later stages of the project as shown by Luiten [1994].

## *5.4.3 The Discipline View Model Paradigm*

The next paradigm that is discussed is called the Discipline View Model paradigm. Elaborations of this paradigm can be found in e.g. [Nederveen, van & Tolman 1992] [Rosenman 1993] [Amor and Hosking 1993] [Tolman 1994a] [Bakkeren 1997] and [Nederveen, van 1993]. The last reference is a paper in which the basic concepts are elaborated and discussed in detail; the paper is attached to this thesis as Appendix A. Below the main characteristics are summarized and some variations in elaboration are discussed.

The Discipline View Approach is characterized by the following:

- The information requirements of a design discipline are defined in a *discipline-specific model.* Such a model will be called a *View Model.*
- The *interfaces* between models can be more complex than one-to-one mappings or subtype-supertype mappings. Communication between design disciplines implies conversions in which mechanisms such as transformation, idealization and filtering take place. In the discipline view model approach, the interfaces between models support such mechanisms. Or, as Amor and Hosking put it: the model interfaces are defined as *functional abstractions* rather than as subset-relationships.
- A *kernel model*, as commonly used in STEP-related work, is still useful. As in STEP or any other standard, a kernel model limits the number of interfaces between models, when the number of models becomes big: if there are n models, then one will need n*(n-1) interfaces without a kernel model, but only 2*n interfaces with a kernel model.

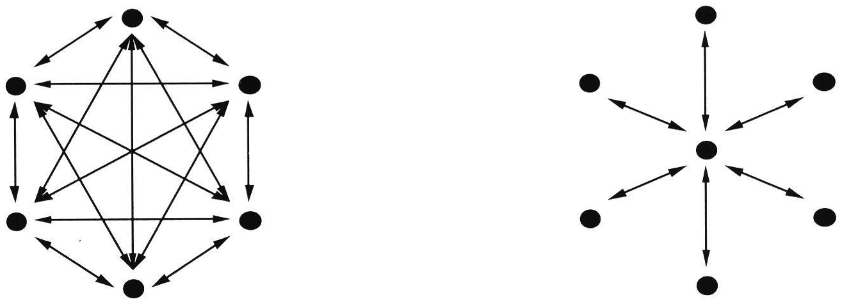


*Figure 5.2 Without a neutral- or core model, the number of interfaces tends to explode. With a neutral model, the number of interfaces remains manageable.*

The basic model structure of the discipline view model approach, resulting from the observations above, is shown in Figure 5.3.

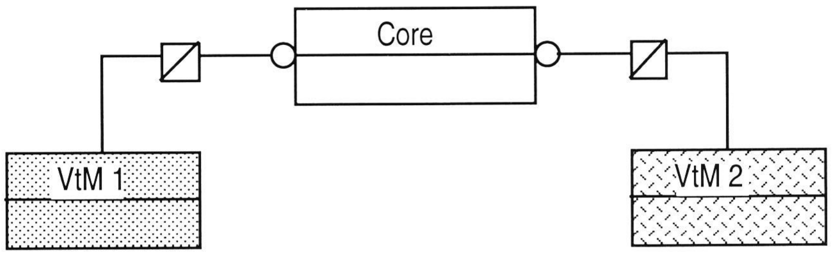


*Figure 5.3. Basic model structure of the Discipline View Model Approach: discipline-specific information is modelled in View type Models; communication takes place via a kernel model and complex interfaces.*

An important advantage of the discipline view model approach is that it allows model extensions without restrictions, in other words it allows “concurrent” model development (see [Alberts & Dikker, 1994]. This is especially convenient for organizations that want to formulate a bottom-up strategy for the development of standardized building product models (as adopted in the Dutch NOBI-project in 1993).

However, too much of a bottom-up approach is dangerous, since there is a significant risk that the resulting models are very difficult to integrate, because the model interfaces are getting very complex.

## *Evaluation*

The Discipline View Model paradigm can be seen as a rather generic approach that can be elaborated in different ways. The common notion is that different disciplines must be supported with specific Discipline View models.

Within the STEP methodology, it is very difficult to realize the Discipline View Model paradigm[7], and this is one of the reasons that the STEP standard falls short in the support of the building and construction industry.

But also without STEP, it has turned out to be very difficult to implement the Discipline View Model paradigm. Although the ATLAS project used a slightly different approach, see below, this project has shown that the most difficult part lies in the complexity of the interfaces. Because discipline models can be structured in fundamentally different ways, the interfaces have to bridge enormous gaps.

Nevertheless, as a design and modelling concept, the Discipline View Model paradigm proved to be a sound approach. And as such, the paradigm can be regarded as yet another "essential classic", comparable to Functional Unit/Technical Solution.

## *The Round Table Paradigm*

This paradigm has been developed by Tolman and demonstrated in the European ATLAS-project [ATLAS 1993] (see also 5.5.1). Basically the idea is to focus directly on the communication between representatives of all the different disciplines involved and to structure and model that communication logically. So in this paradigm the objective becomes to find out what representatives of different disciplines (sitting together at a round table) are talking about with each other (and in which words)[8].

The key difference with the Discipline View Approach described in the previous section, is that the Round Table Approach takes the communication between disciplines (the "Round Table") as a starting point, not the individual discipline views.

An analysis of this communication process shows that interdisciplinary communication is rather superficial. Common notions are mainly restricted to those that globally describe the artefact, i.e. Room, Floor, Wall, etc. or the process, i.e. Task, Site, Area, Cost and Time (all with attributes for identification). The most detailed communication

[7] Since Interoperability between different Application Protocols is not supported

[8] Imagine them sitting around a round table and listen to what they are talking about.

occurs on the interfaces between two disciplines. Examples are: pipes, cables, ducts and wires that penetrate walls and floors, weight of equipment that has to be carried by floors, etc.

### *Evaluation*

A key advantage of the Round Table paradigm is that it concentrates on the essential basic concepts that are communicated. In other words, this paradigm tries to capture the shared ontology of different disciplines. As such, the round table paradigm offers a more focused approach than the Discipline View Model paradigm discussed above resulting in a rather small core model, or set of core models.

The weakness of the Round Table paradigm is on the implementation side. As with the Discipline View Model paradigm, it is very difficult to define and implement and maintain the interfaces needed to exchange the information "at the round table". Though ATLAS showed that the concept works (see 5.5.1), it is not really implementable in an adhocracy as STEP.

## *5.4.4 Process Oriented Paradigms*

A common characteristic of the paradigms discussed above is they are basically data-oriented. The paradigms focus on the data that must be communicated. Although the proper definition of the data is a prerequisite for successful meaningful communication, it is found insufficient by several authors.

In this section two approaches are discussed in which a more process-oriented paradigm is advocated.

### *Transaction-Based Communication*

The first approach discussed here is the approach taken in the Dutch VISI-project [VISI 1999]. This approach is based on work by Dietz [1996] and starts from the notion that the *transaction* is the basis for communication.

A transaction is seen as an event in which:

- A communication *process* takes place,
- Several *roles* are involved (whereas participants can have different roles in different transactions),

- A specific *information content* is communicated.

In Fig 5.4 the approach taken by VISI is illustrated. In the middle, a transaction is defined called “Delivery of Detail Design”. This transaction takes place between two roles of participants: “Order Detail Design” and “Make Detail Design”.

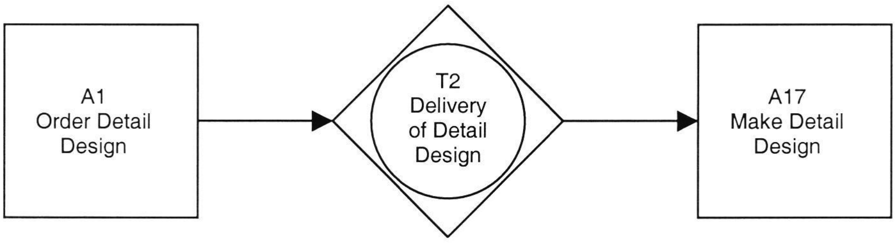


*Fig 5.4 The main elements of the transaction approach followed in the VISI project [1999]*

Next, sub-steps of the transaction can be specified (e.g.: “give order”, “confirm order”, “execute order”, “finalization call”, “acceptance”). Also roles can be elaborated, by specifying messages sent, messages received, responsibilities, etc.

Based on the notion described above, the VISI-project has elaborated a so-called VISI-framework. In this framework the organization, processes and data of a construction project are modelled in terms of transactions and roles.

The goal of the VISI framework is eventually to serve as a standard for electronic communication in building and construction in the Netherlands. For that reason the project intends to proceed with implementation and testing of the framework in a number of pilot projects, in close co-operation with software companies. In a few years this should lead to the establishment of the framework as a standard.

## *Evaluation*

One disadvantage of the approach is that it is limited to the situation in the Netherlands only. Communication over the boarders is a complication not yet in scope of the project. However the ideas can be used also for the development of international communication standards.

Furthermore the VISI effort seems to be yet another long term top-down effort, comparable to the many STEP efforts as well as national efforts such as the Dutch BIM-project [IOP-Bouw 1989]. The dangers of such an approach are clear: it takes a long time until practical results can be expected, there is a serious risk that the framework will become too complex for practice and/or that key participants (e.g. contractors or software vendors) will not accept the proposed standard.

On the other hand, the transaction approach, as advocated in VISI, might lead to a significant improvement compared to the data-oriented approaches used elsewhere. At least it is a refreshing approach. Moreover, it is very likely that the developers are fully aware of the dangers discussed above. Very positive is that they have worked on the commitment of many key players in the Dutch building construction industry from the very start.

The next stage of the project will prove whether they will be able to cope with the other dangers of the top-down character of the project. But the project deserves a chance.

## *Task Support Using Project Windows*

At the end of the COMBINE 1 project [Augenbroe 1993], one of the conclusions of the project team was that their STEP-based information architecture had serious limitations with respect to the support of the dynamic characteristics of design. Moreover, this was found a general problem of the STEP-related projects at the time. For that reason, one of the objectives of the COMBINE 2 project was a better support of the dynamics of design.

In COMBINE 2 (1992-1995), this objective has led to the development of an approach for design task support using so-called Project Windows. Project Windows can be regarded as small subsets of the building process in which a few design tasks or functions of a few design actors are modelled, including the data that is exchanged between these tasks. This resulted in prototypes in which data is exchanged between design participants in a STEP-conformant way, but controlled by a Project Window-specific process definition.

The basic idea is shown in Fig 5.5. In this figure, a Project Window is shown with three Design Tasks. Between these tasks, transactions take place: output of one task forms input of the next task, etc. Other options that are not shown, are feedback loops, alternative scenarios (e.g. sometimes certain tasks do not need to be executed), etc.

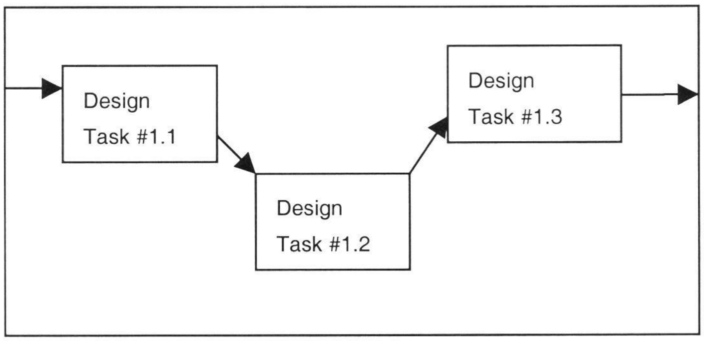


*Fig 5.5 The Project Window approach as developed in COMBINE*

More details on Project Windows can be found in [Augenbroe 1995] and [Augenbroe et.al. 1998].

## *Evaluation*

The Project Window paradigm as developed in COMBINE 2 is potentially a very valuable contribution for electronic communication in building and construction. Project Window-controlled communication could very well be based on the common building concepts such as walls and columns, as was argued at the beginning of this section. Furthermore, the addition of process control seems to be a way to support the dynamics of design in a controlled way.

However, it is not very likely that this approach will result in short term improvements in building practice. First of all, the paradigm uses STEP-technology for data exchange, and thus has to deal with the STEP-issues discussed before. Secondly, the modelling of design tasks is yet another area of many research issues. Some of these issues have been subject of the VEGA-project (see 4.2.4) in which integration of STEP-technology and workflow-management is one of the project objectives.

It was stated earlier that the STEP-approach is probably too difficult for short-term solutions for electronic communication in building and construction. The Project Window paradigm provides some promising advantages over STEP, but is likely to be even more difficult and further away from short-term benefits for the building industry.

On the other hand, the idea of Project Windows could possibly be a good starting point for the use of workflow management systems. In this way, a rather abstract research concept might be able to reach daily practice quickly, by using commercial software for project management tasks.

### 5.4.5 Small Model Approaches

#### The Minimal Approach

An often-stated criticism towards both STEP models (as developed as part of Application Protocols) and STEP-related data models, points at the size of the models. STEP models tend to become very big, with hundreds of entities and attributes, and thus very difficult to manage and maintain.

This has led to proposals for small and manageable "core models", to be developed using a "minimal approach", as proposed by De Vries [1991] and Tarandi [1998].

Basically the idea is simple: delegate semantics to a Classification and Coding system like SfB, or BSAB. Figure 5.6 illustrates the idea.

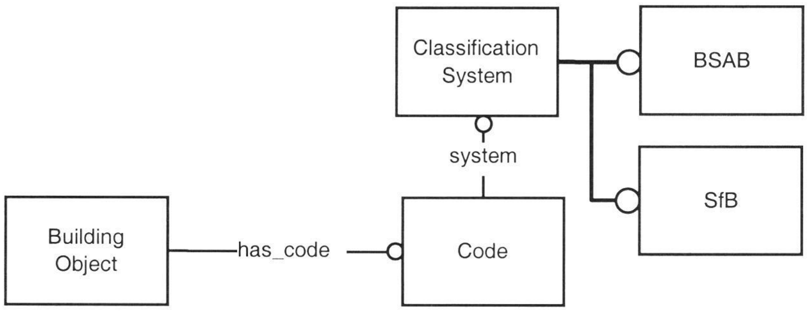


*Figure 5.6 BuildingObjects are not specialized in the model (notions like wall, floor, beam are not used). A classification code determines what kind of object it is.*

#### Evaluation

Using references to classification systems has been tried before, in some cases with great success (see [Tarandi 1998]). The problems with this approach are: (1) that it is not possible to follow this approach on an international scale, because different countries use different classification systems, (2) mappings between classification systems are not possible and (3) expressing rules is not possible, or at least extremely difficult and unreadable.

Incompatible classification and coding systems (incompatible object definitions) will also slow down the development of interdisciplinary communication development in the coming years. The aecXML initiative is aiming at a minimal model approach. It hopes to use the USA based Masterformat and Uniformat classification systems for its object definitions. As this approach is not applicable in Europe where each country has its own incompatible set of classification and coding systems, the European project E-Construct is tackling the problem of 'neutral' object definition as part of its BCML vocabulary development. Another promising development is the co-operation of European classification groups in ICIS, in which participants from the UK, Norway and the Netherlands work on a mapping between their national terminologies.

## 5.5 Lessons from Earlier Projects

### *5.5.1 ATLAS*

ATLAS aimed at the development of standards and tools for interdisciplinary communication and inter-sector communication [ATLAS 1993, Tolman 1994a, Tolman 1994b, Van Nederveen 1994]. The sectors involved where Process Plant and Building-Construction. The idea was to provide a mechanism that supported the design and planning of technical buildings, where a team of building and civil engineers co-operated with a team of process plant engineers.

Within B-C the following disciplines were involved:

- Architects
- Structural engineers
- HVAC engineers
- Project managers
- Cost estimators
- Process planners

And within Process Plants the following disciplines were taken into account:

- Process engineers
- Piping engineers
- Instrument engineers

In order to solve the communication problem ATLAS developed a layered model architecture (see Figure 5.7) using core models according to the round table paradigm

(5.4.3). The top model was called the LSE Project type Model (LSE PtM). The LSE PtM provided the functionality required to communicate between CAxx systems of different sectors of the LSE industry. One level below there were two sector specific models, the B PtM (Building Project type Model) and the PP PtM (Process Plant Project type Model).

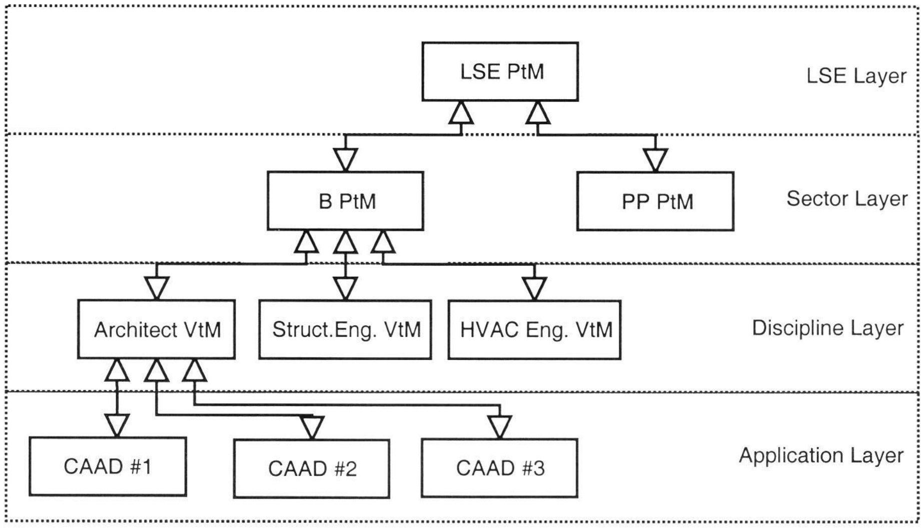


*Figure 5.7 ATLAS model architecture. Communication between application systems of one discipline is supported by a discipline specific View type Model (VtM). Communication between disciplines of one sector is supported by a sector specific Project type Model (B PtM and PP PtM). Communication between different sectors is supported by the LSE PtM. The figure shows one branch of the model tree.*

The LSE PtM distinguishes four types of objects: *Product* (Building, Wall,..), *Resource* (Equipment, Formwork, Crew,..), *Process* (Activity, Task,..) and *Control* (Contract, Drawing, Planning,..).

The core statement used in ATLAS is:

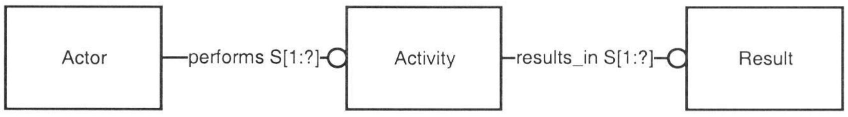


*Figure 5.8 Actors perform Activities that result in Results.*

*Results* were divided in Product Results that contribute to the artefact, Resource Results that are used as resources in downstream processes (like formwork) and Control Results (i.e. a set of drawings).

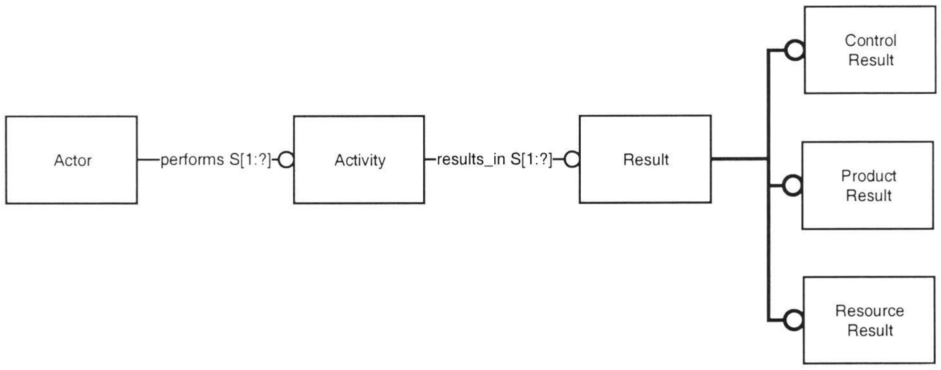


*Figure 5.9 Several types of Results.*

Subsequently the different groups of objects were detailed and relations between the different groups were modelled. Figure 5.10 shows the product objects described in the LSE PtM. Product objects are grouped into objects that belong to the same system (see chapter 3), i.e. space system, structural system, separation system and flow system (which specializes further into HVAC etc.). Also the relations between these objects (called *interrelations*) can be used in the communication. Product objects in the LSE model retained some of the properties that were of general interest like weight and heat radiation of HVAC-flow system.

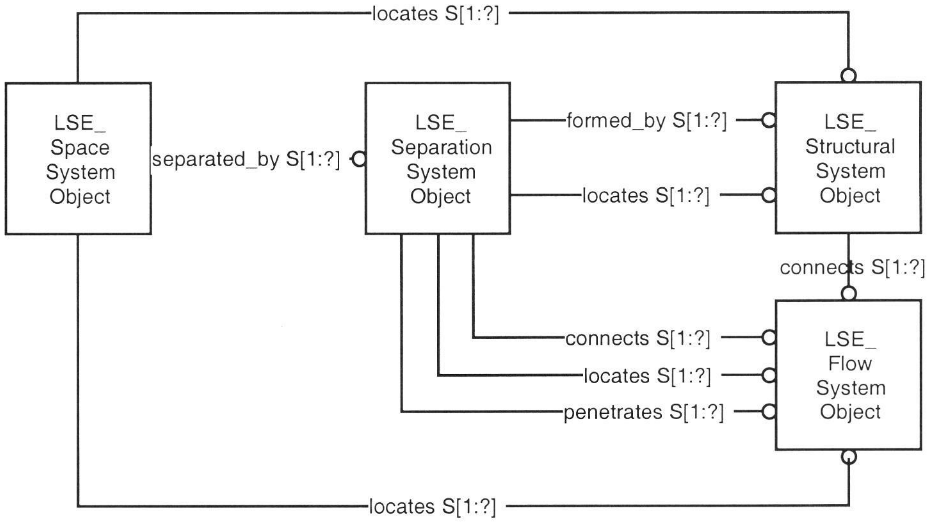


*Figure 5.10 The 4 basic Product Objects of the LSE PtM and their relations. This are the notions that can be used in the communication between sectors (here Building and Process Plants). Sector specific PtMs are specializations of this model. This means that a sector PtM can be instantiated on two levels at the same time. A Turbine in the PP PtM becomes a LSE_FlowSystemObject in the LSE_PtM (with weight, heat radiation, location, etc.).*

Communication between CAxx systems of different sectors mainly focuses on the *interrelations*, i.e. which wall or floor is penetrated by which pipe, or which process equipment is connected to which floor. This type of simple communication was already enough to support the concurrent design of a building and the process plant equipment housed in the building. Changes made by one of the Building partners immediately could be communicated to the (CAxx systems of the) Process Plant designers and vice versa.

## *Evaluation*

In the final demonstration ATLAS showed that meaningful information interchange can be developed using the concept of layered models, but at a price. The price is that a large amount of effort goes into the development and maintenance of the translators. In fact the translators become more or less integral part of the standard.

Another lesson from ATLAS is that the development of a comprehensive set of models is a task that requires a dedicated team of experts that are willing to follow a common set of rules.

For example, it turned out that the writing of interface protocols was not only a huge effort because of fundamental differences between the different discipline view models. An even bigger problem was to convert different modelling approaches, different shape description methods etc. Because ATLAS was a closed environment with strong project management and sufficient financing the project goals could be met.

Standardization in ISO STEP however is a democracy, or maybe even an adhocracy, or anarchy. Definitely not an environment that is able to follow the ATLAS approach with any chance of success. A top down ATLAS-like approach is not the solution for the fragmented Building and Construction industry.

## 5.5.2 The Initial BCCM

The Building Construction Core Model (BCCM [ISO 1994], developed as STEP part 106, initially followed the ATLAS modelling method. The lessons learned from ATLAS were taken at heart. No multi-level layered modelling, but a specialization hierarchy was the goal.

Figure 5.11 shows the root statement that divided project objects in 4 distinct classes. Each class was detailed further, more or less down to the same level of detail. Figure 5.12 shows a further detailing of the Product Objects and the implementation of the life cycle dimension (required, proposed, realized).

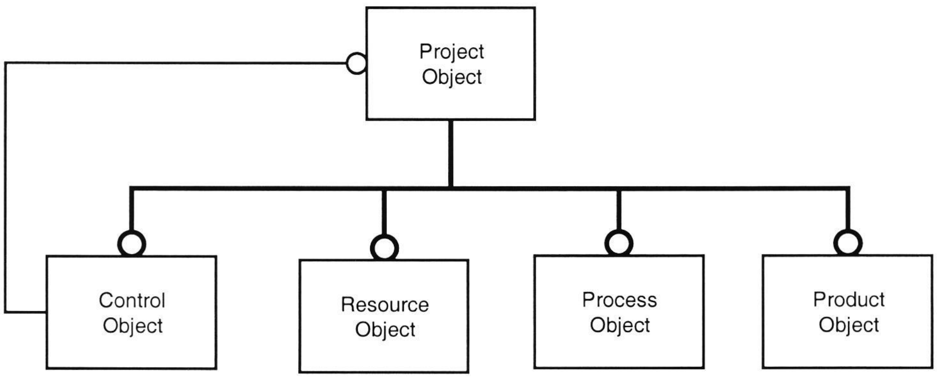


*Figure 5.11 Core of the initial BCCM. Note the multi-level control structure and the fact that the model strongly resembles the IDEF-0 paradigm, except the implicit fact that here processes may result in other processes.*

The BCCM tried to define a model that supports communication in a *project*, and includes a minimal set of notions that can be used for the job. Unfortunately also the BCCM was not a really small model with more than 200 objects.

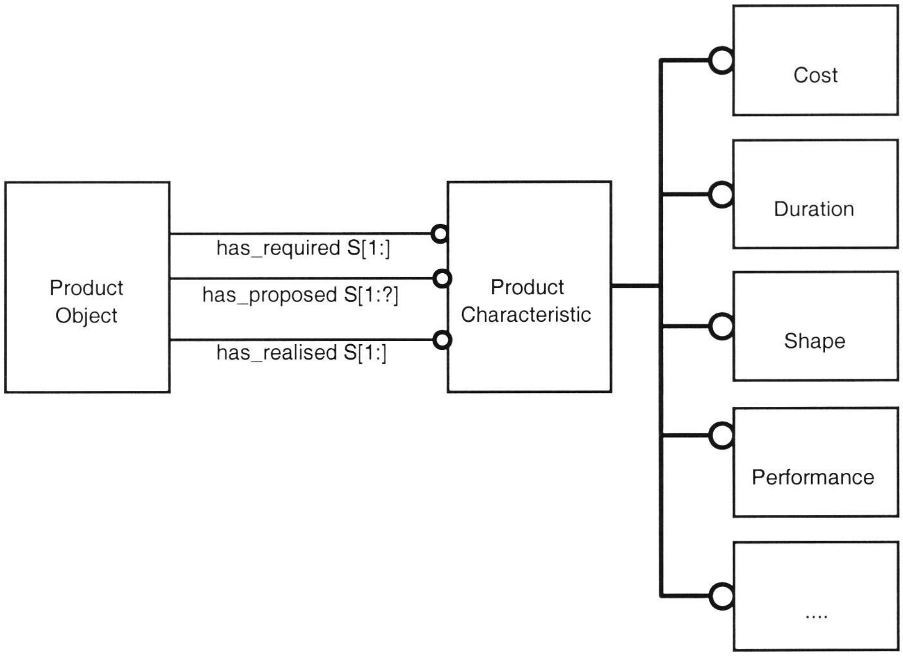


*Figure 5.12. Illustration of the implementation of the life cycle dimension.*

After two years the team had to stop the effort due to lack of financing. From that time onwards the Industry Foundation Classes (see below) were replacing the initial BCCM.

## *Evaluation*

The interesting contributions of the BCCM were (1) the attempt to model 4 groups of project related objects in harmony, and (2) the explicit implementation of the life cycle dimension (i.e. requirements, design decisions and final results). The BCCM supported communication about every important aspect of a construction project, though not in much detail. Details were left to the specialists.

Also important is the lesson that the Building-Construction industry is extremely fragmented and weak as it comes to the development of international standards for interdisciplinary communication [Tolman 1999]. Though the subject is of course of major importance to the industry as a whole (several experts estimate that a 30% decrease in overall project costs is easily obtainable), nobody is willing to finance the development. Some of the most important reasons for the reluctance to standardization are: (1) national governments – one of the major players in the sector – are only willing to finance national projects, (2) large international client organizations are unable to find each other (3) the ISO STEP standardization process is ineffective, has no leadership and has no power, and (4) BC companies fear a loss of income if the construction process is really brought up to date.

## 5.6 Lessons from the Latest Efforts

In this section three important PDT-projects for building and construction are discussed briefly:

- The development of Industry Foundation Classes (IFCs) by the International Alliance for Interoperability (IAI),
- The EC-project CONCUR,
- The LexiCon-project.

### *5.6.1 IAI-IFC*

The initiative of a group of companies that together form the International Alliance for Interoperability (IAI) is probably the most practical development that currently takes place is. The IAI promotes the development and implementation of the Industry Foundation Classes, a STEP-like format for BC information sharing and exchange in the building design stage. The IFCs are a follow up of the ATLAS-project, the STEP BCCM development and the EU COMBI-project [Scherer 1994].

In order to give an impression of the approach and structure of the IFCs, a brief analysis of the IFC Core model is presented below. The analysis is based on version 2.0 of the IFC Core model. At the time of writing, this was the latest public version. Version 3 is also finished, but not published, while currently version 4 is under development.

The most important characteristics of the IFC model can be found in Figure 5.11.

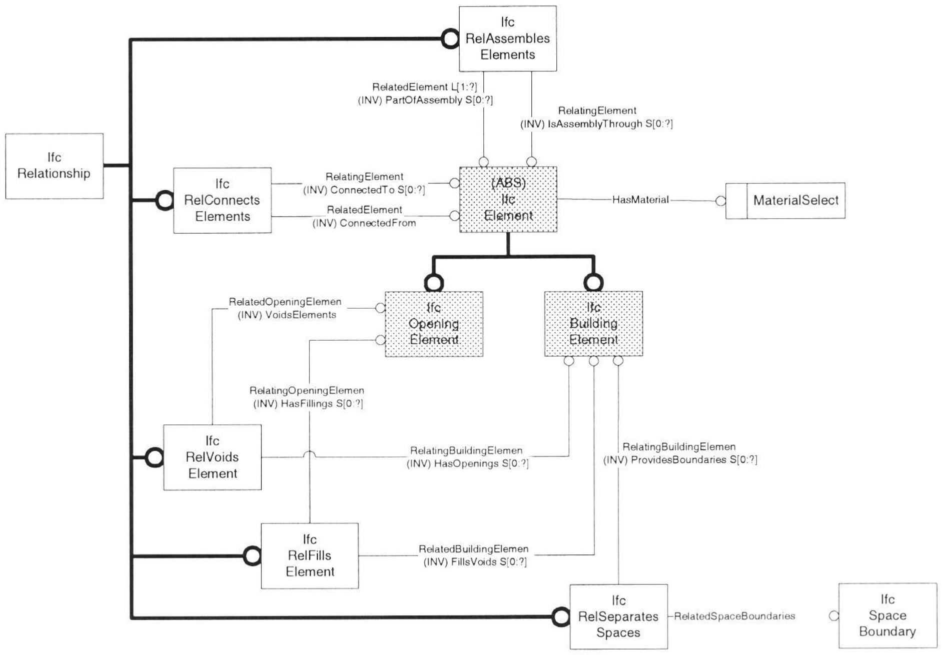


*Fig 5.11. EXPRESS-G schema of the main model constructs of the IFC Core model, used to describe building elements such as spaces, walls, columns, windows and doors.*

In this schema, the following characteristics of the IFC Core model can be recognized:

## *Model Architecture*

The model architecture of the IFC Core model is a lot simpler than the ATLAS architecture as shown in Fig 5.7. The IFC Core model has constructs for specialization (e.g. the subtypes of IfcElement and IfcRelationship) and decomposition ("contains").

But the model does not have distinct layers for generic data, sector specific data, discipline-specific data and application-specific data as in ATLAS. This results in a simple structure with a flatter (less-hierarchical) model.

## *Objectified Relationships*

For relational structures, so-called objectified relationships are used. In other words: relationships between entities are described using other entities, elsewhere sometimes called associations. Here, these entities are subtypes of the generic entity IfcRelationship.

The use of objectified relationships is not new or unique. In several other models objectified relationships have been used, for example in AP 221, and earlier in the PISA project. The use of objectified relationships origins from the limitations of the EXPRESS language in terms of object-orientation (i.e. inheritance of relationships characteristics).

The advantage of objectified relationships is that relationships can be described in a very accurate way. For example:

- both directions of a relationship can be defined explicitly, including constraints on the relationship,
- specialisation structures of relationships can be added.

A disadvantage of objectified relationships is that the resulting model becomes a lot more complex than in the case of "simple" relationships.

### *Properties and Property Sets*

Properties and property sets are not visible in Fig 5.11. These are described in separate tables. In this way, the entity-level data structures are not "spoilt" with the lower level property information. However, it shows once more that the IFC Core model is far from simple & small, as the model was meant to be.

### *Evaluation*

Probably the strongest point of this development is that the IAI is not a democracy, but a group of stakeholders with a common goal.

The strength of the IFCs is that IFC-Objects are mainly classifying common names, which are disconnected from the attributes. This means that much of the complications have gone. On the other hand however it introduces another problem, i.e. incompatible definition of attributes.

The development history of the IFCs shows some typical features of the big model vs. small model discussion. The layered ATLAS model structure was found too complex, leading to a flat model structure in the BCCM. Then the number of entities was found too big, and a modular approach was introduced, similar to the approach initially taken in ATLAS.

But the real issue of big models vs. small models does not seem to be solved. The danger of big models is clear: difficult to keep consistent, complex interfaces, and very difficult to maintain. Approaches such as layering and modularization can provide some help in consistency and maintenance, but cannot take away all dangers, as was learned in the ATLAS-project.

The danger of small models is also clear: small generic models tend to become abstract and meaningless, resulting in non-semantic exchange. Approaches with classifications or parts libraries (see 5.5.) can provide the semantics that are needed. But again, this does not solve the whole problem, since part of the problem (i.e. the semantic definition of concepts) is now moved to the classification or library domain. Also, the attractive simplicity of small models has its price: in most cases the relationships between objects are only modelled on a very abstract and meaningless level.

*Conclusions*

The observations presented above show that the IFC Core model effort has not (yet) succeeded in the development of a small, simple, easy to implement & to use model. This confirms the hypothesis that a true simple, ready-to-use data model may require a more pragmatic, bottom up approach.

If the current IFC approach is indeed the way to go can only be answered in time. It is not impossible that the same problems as encountered in ATLAS (mainly following from the mappings required) will also show up here. Maybe managing a flat model with a few thousand words (as done in the LexiCon, see 5.6.3) is not as difficult as the IFC-team presumes, according to [Junge and Liebich 1998].

Another problem with the IFCs is that the development cycles are going much to fast. IFC1.5 is not yet available on the market, but IFC2.0 is already available. And IFC3.0 and 4.0 are under development. These fast development cycles are not what the vendor community wants.

### *5.6.2 CONCUR*

A project that involves aspects of interdisciplinary communication is the Brite-Euram CONCUR-project, which started in 1997. CONCUR, like ATLAS, has to bridge two sectors. In this case Building-Construction and Power Plants. The initial goal of CONCUR was (1) to bridge several islands of integration, such as IAI-IFC, AP221, PLib and CIM-Steel and (2) to integrate three project life cycle stages: inception, design and specification. The idea was to develop an ontology of Objects Of Interest (OOIs) that describe the three stages in detail together with a set of Characteristics, or Properties following from a number of Aspects of interest.

Although at the time of writing of this text the CONCUR project is not finished yet, it becomes already clear that the initial goals will not be met and that the project will mainly focus on integration around the IAI-IFC, i.e. concept and scheme design of buildings only.

Another problem with CONCUR is that IFC integration is implemented around the IFC *shape* entities and not by following a true product modelling approach. This has been a consequence of the initial project objective to focus on integration of *current* CAD-systems (AutoCAD Architectural Desktop, Allplan, ArchiCAD).

This deviation of the true PDT spirit makes it hard if not impossible to integrate with other 'islands of integration' like the Epistle island and other developments in the Power Plant industry and will undoubtedly colour the results of CONCUR.

On the other hand the project really contributes to the integration of a lot of applications around the IFC core, which, for a Brite-Euram project is definitely a positive thing.

### *5.6.3 LexiCon*

An interesting new development, proposed by Woestenenk [Woestenenk 1999] in CONCUR, is called the LexiCon. The LexiCon is a collection of words (objects) that describes artefacts such as buildings in great detail (down to the specification level). The structuring mechanism has, with slight adaptations, been borrowed from the General AEC Reference Model (GARM) [Gielingh 1988]. A Functional Concept (like wall) can be realized by a number of Solution Concepts (cavity wall, etc.). Each Solution Concept is made up of a set of lower level Functional Concepts, for example, cavity wall is made up of Function Concepts like: cavity, wall tie, brick, etc.

Also interesting is the fact that objects in the LexiCon can have different *Roles*, thus incorporating a mechanism to deal with objects that play a role in different systems.

In order to give an impression of the approach and structure of the LexiCon, a brief analysis of the LexiCon model is presented below.

At the time of writing, there was little data modelling work available. The LexiCon report only contains a single, simplified EXPRESS-G diagram [Woestenenk 1999, p 11]. However, from the accompanying text, a picture can be formed of what is meant in LexiCon. This has resulted in Figure 5.12.

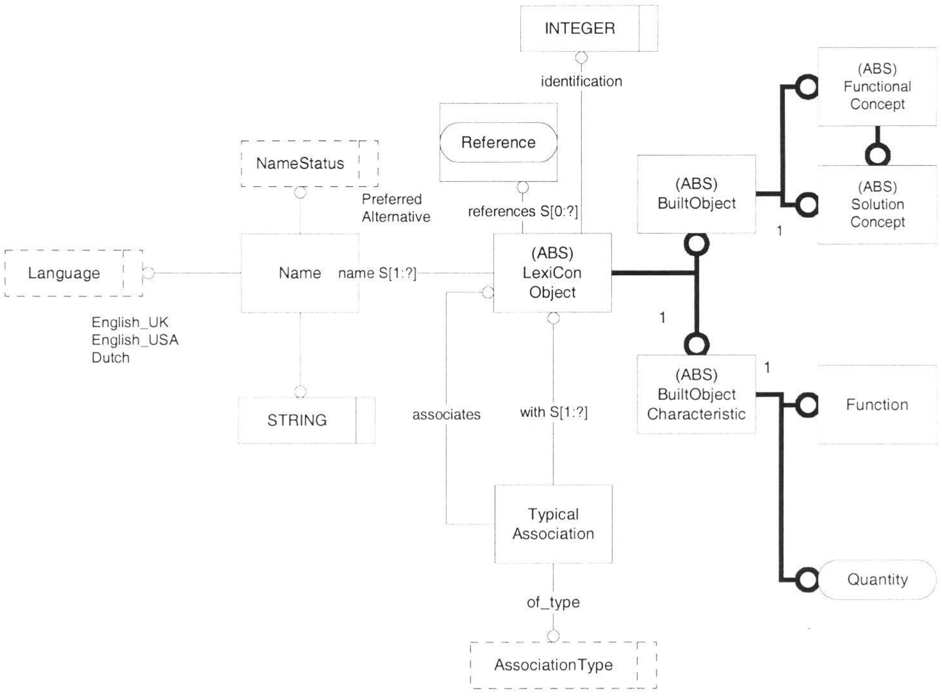


*Fig 5.12 Overview of the main constructs proposed in LexiCon*

In this schema, the following characteristics of the LexiCon can be recognized:

## *Relationship with External Classifications and External Data Models*

The LexiCon provides a relationship with external classifications or data models using the relationship LexiCon Object – references S[0:?] – Reference. This Reference could be about anything: the SfB classification system (in any of its flavours), the ancient ISO 6007 on Building Terminology, ISO 31 on Quantities and Units, dedicated building parts libraries, or STEP Application Protocols such as AP 221.

Of course, this construct will need a lot of elaboration, but the general approach is valuable.

### *PDT Concepts*

The rest of the schema shows the basic PDT concepts that are used in LexiCon:

- The entities Functional Concept and Solution Concept, based on the GARM concepts, see 5.4.2,
- Functional Characteristics and Quantity Characteristics,
- A general objectified relationship called Typical Association, with an AssociationType.

Furthermore, the LexiCon has an elaborated construct for the naming of objects, including names in different languages; see the left area of Fig 5.12.

### *Evaluation*

The LexiCon can be seen as an attempt to link between PDT, classification systems and electronic libraries. As such it is a valuable contribution to product data technology. The PDT part of the LexiCon describes some essential PDT concepts in a nice small data structure.

Yet two points of attention must be made:

- The LexiCon is at an early development stage; a lot of work still has to be done, specially the filling;
- It is questionable whether the nice & small abstract data structure will survive when the model is going to be tested in a real life project.

### *Conclusion*

Work on the LexiCon has only recently started. In the coming years the LexiCon will be filled with thousands of words and made available (in electronic form) as a new kind of classification system. In this case with a classification system that supports unique classification of objects. The Lexicon also plays a role in the recently started European E-Construct project where the emphasis will be to solve the multilingual and multi-classification problem that the European Building and Construction industry still faces.

Hopefully in the future the LexiCon can help to keep the models small without loss of too much meaning.

### 5.6.4 Basic Semantic Register

Another initiative with a similar purpose, though mainly focusing on E-Commerce, has been taken by ISO/TC154/WG1.

The BSR is an official ISO data register for use by designers, implementers and users of information systems in a manner which will allow systems development to move from a closed to an open multilingual environment, especially for use in domestic and international electronic communication including electronic commerce and EDI.

The purpose of the BSR is to provide an internationally agreed register of multilingual data concepts, semantic units (SU), with its technical infrastructure. This will provide storage, maintenance and distribution facilities for reference data about semantic units and their links (bridges) with operational directories. The semantic units will be built from semantic components, which can be considered as building blocks. It should be noted that the term "directories" also includes "repositories". The BSRs principle function is to provide data in multiple languages, which have been developed in a consistent, unambiguous manner according to international standards. This development will be carried out in partnership with the organizations that have responsibility for it.

## 5.7 Conclusions

Looking back at what has been tried it seems fair to say that all the earlier efforts were too much top down and too complicated. The approach followed in these projects resulted in large and complicated models that were difficult to develop, implement and (above all) maintain. Consequently none of the approaches had too much success outside their developer's circle. The ATLAS project showed that a layered model approach can be successful, but only if developed by a team of modellers that is well managed. Unfortunately in Building and Construction chaos is the common denominator, at least on the international market place. In such an environment ISO STEP/BC is too democratic, too poor, and much too weak to enforce a common set of rules, which is clearly demonstrated by the other sectors as discussed in chapter 4. The idea that ISO STEP will develop standards that will later be implemented by vendors does not seem to work in BC. Too much fragmentation and lack of rich, dedicated market leaders makes it extremely difficult to come up with something useful for BC.

Therefore the latest efforts tried to find another, more realistic approach to solve the problem. It has become very clear that efforts on electronic communication in building and construction only have a chance if the model structure is kept very simple, flexible, and easy to implement and to maintain.

This means that a selection must be made of essential concepts that must be supported. It is obvious that the rich palette of concepts that have come across in building product modelling over the years is impossible to implement in a short-term effort.

In fact, the latest efforts discussed in 5.6 can be seen as attempts to capture such an essential subset of concepts that is needed for electronic communication in building and construction on a semantic level.

From the latest efforts it seems that the IAI-IFC has the best backup, although even this should not be exaggerated as the amount of financial support is still very limited, i.e. only a few percent of the marketing budgets of the contributing end users. This shows that BC still does not see improved electronic interdisciplinary communication as a major solution for its problems (inferior price/performance). At the down side it should be noted that the IAI-IFC is not only a solution, but also partly a contribution to the problem, as it is another island of integration.

From the other efforts the Minimal Model Approach is quite interesting, provided that the Classification people can provide us in the future with a “neutral” classification system (perhaps better called a neutral identification system, because semantic *identification* is the main issue). Also the development of the LexiCon is a good idea, which might serve the industry in the future.

Finally the latest attempts to start electronic communication with E-Commerce might be a good idea. At least here the benefits are clear both for the information providers and information users. In this context the new EU-project E-Construct (E-Commerce for the Construction Industry) is an interesting initiative.

For now it seems reasonable to say that 20 years of R&D in Product Data Technology did not bring the Building-Construction industry what it needs: electronic interdisciplinary communication. The most important reasons are that BC is too fragmented and too weak to follow any of the more successful approaches applied by the other industrial sectors. It seems that a much more loose and bottom-up approach is needed. All the approaches that need international standards to be developed and implemented in a large amount of CAxx systems are probably bound to fail. Too many obstacles and risks threaten this kind of approach.

Turning away from international solutions to national approaches, like is done in the CORENET project in Singapore, is also not a solution for the international marketplace. Such an approach easily results in less then optimal ICT usage, because the main ICT vendors coming from abroad are not likely to support national standards.

Where does that leave the Building and Construction industry? One answer is: in search of an approach that suits the needs and culture of our industry.

# Reformulation of the Research Question

*This chapter summarizes the constraints for new solutions for interdisciplinary communication and reformulates the research question.*

## 6.1 Introduction

In chapter 1, the initial research question of this thesis was formulated as follows: "How to use state of the art Information and Communication Technology, to improve interdisciplinary communication in large-scale building and construction projects?".

In the previous chapters a number of developments on (electronic) communication in practice and in research have been reviewed, many of which are based on Product Data Technology (PDT).

For Building and Construction also a Product Data Technology (PDT) approach is required, but not a top down approach starting with the development of international electronic communication standards. That road is much too long and too winding, and does not bring the required results.

If possible and feasible another approach should be tried out. This chapter tries to reformulate the research question.

## 6.2 Weighing Up the Pros and Cons

In order to find a possible and feasible approach, the strengths and weaknesses of the existing approaches are assessed. This is done in order to stimulate the strong points and reject the weak points of the existing approaches.

Strong points that should be stimulated are:

- The approach should be semantic; the idea of meaningful product models is too valuable to lose.
- The approach should be "neutral"; i.e. independent of software applications, vendors, building participants and standardization efforts.
- If possible the approach should first serve the needs of the most powerful participants in the project. Earlier approaches mainly started from Design/ Engineering. However designers and engineers are not powerful enough to make a fist.

Weak points that should be discarded are:

- No time consuming international standardization effort should be required.
- No top down, theoretical approach should be needed.
- The approach should allow the project partners to start whenever and wherever they want.
- Interfacing with existing CAxx systems should be possible.
- Shape description is important, but it does not have to be treated as a key element of interdisciplinary communication; interfaces with CAD-systems should be able to take care of shape description and visualization.

## 6.3 Conclusion

The reformulated research question is: "How to develop an approach for interdisciplinary communication in large-scale building and construction projects that is neutral, pragmatic and bottom up, primarily serving the needs of (one of) the most influential participants in a project?"

As a consequence of this conclusion the Project Manager was selected as the prime target group for this research. In order to serve the Project Manager's needs the solution provided should focus on interface management, risk, and information management in general and should not require large investments in tools and learning.

# The Object Tree Approach

*This chapter develops an answer to the reformulated research question of chapter 6. The first part of this chapter discusses modelling aspects; the second part discusses implementation aspects. The Object Tree approach described in this chapter is tested in the case study described in chapter 8.*

## 7.1 Introduction

In chapter 5 it was concluded that an eventual success of the existing Building and Construction PDT approaches in terms of practical applications in the coming years is at least doubtful. A more bottom up (but still *semantic*) approach is required. This chapter proposes a new approach, termed the Object Tree approach. Below a number of important aspects about the Object Tree approach are explained in more detail.

## 7.2 Modelling Object Trees

An Object Tree (OT) is a hierarchy of objects that describes a *particular* project in some detail, resulting in an artefact (a product). The level of detail depends on the purpose of the OT.

Starting point of the OT approach is a choice of a particular technical solution for the artefact and the choice of relevant systems and interfaces that define the solution.

While engineering the various systems a large number of 'Objects' become visible. Choosing, dimensioning and realizing these Objects, while satisfying the interface requirements, finally results in the required artefact.

### *7.2.1 Type versus Instance*

An OT is not a *type* model, but an *instance* model. A type model describes a whole class of objects, e.g. the IAI-IFC describes the whole class of buildings. An instance model describes one particular building, i.e. the TNO office building located at the Schoemakerstaat in Delft, The Netherlands. In fact an OT is more or less equivalent to a product model, but without the current emphasis on topology and shape[9]. As discussed in chapter 5, most, if not all, existing PDT-projects focus on type models that describe a whole class of artefacts. For example ATLAS and COMBINE mainly focused on type models for buildings.

The reason for the concentration on type models is the idea that instance models can be more easily derived from type models. So type models are only useful if they come with a tool that can be used for *instantiation* (deriving an instance or product model from a type model). Most often of course a CAD system is used to instantiate the type models.

This shows a weakness in the type modelling approach; nothing is gained by a mere type model; only if the type model is accompanied with an instantiator the model becomes useful. Another weakness is that type models have to be implemented by several vendors on an international level. Consequently the models have to undergo a more or less democratic standardization process. This might easily take years, especially in the fragmented Building and Construction industry.

In contrast, an instance model is made for the direct purpose of the project. Developing an OT is very pragmatic and down to earth. In the next section the Object Tree approach will be described formally.

The main objection to the instance modelling approach is that an instance model has to be developed specifically for the project. In practice this objection is not very important. First there are tools that can help to set up and manage an OT. Second, the problem mainly comes when *shape* is seen as an important issue. Third, simple cut and paste operations can be applied to copy parts of the OT that reappear in more

[9] Shape is seen as one of the lesser important aspects; it is only treated indirectly as will become clear in the rest of this chapter.

places in the model. Fourth, large-scale building and construction projects always have Objects that are specific for the project; those Objects are not supported by the instantiation systems anyway. And fifth, a type model could always be developed afterwards. And this opens a door to "bottom up type modelling": first develop an instance model (or a couple of instance models), and then aim at generalization in a type model.

Finally, the most important advantage of the OT approach for large-scale engineering is the fact that it is a pragmatic, bottom up development that can be started by the project team at any time, without becoming dependent on anybody.

In the following the meta-model of the Object Tree approach will be developed.

## 7.3 Meta-Model of the Object Tree Approach

### *7.3.1 Most Objects are Function Performers*

Most Objects in the Object Tree approach are Function Performers. Exceptions are "Related Objects", which will be defined in 7.3.2. Figure 7.1 shows the idea of Objects as Function Performers. Function Performers have characteristics (for example dimensions) that follow from the systems requirements.

A Function Performer can have a large number of characteristics. Which characteristics are taken into account is to be decided by the project team. In the rest of this chapter a number of characteristics will be looked into.

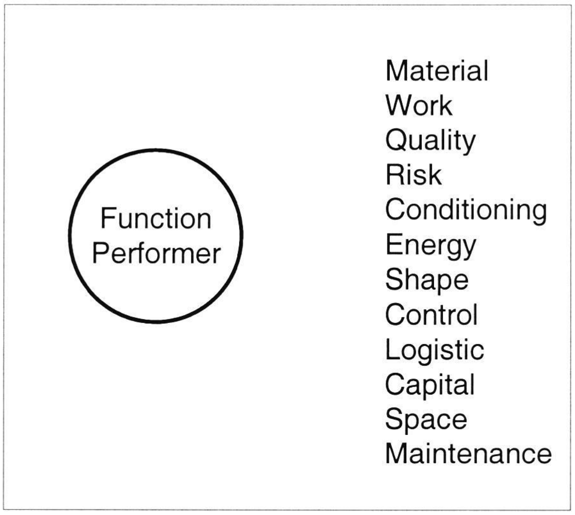


*Figure 7.1. Function Performers have a number of specific characteristics, which follow from the system requirements.*

Figure 7.2 defines the above more formally in Express-G.

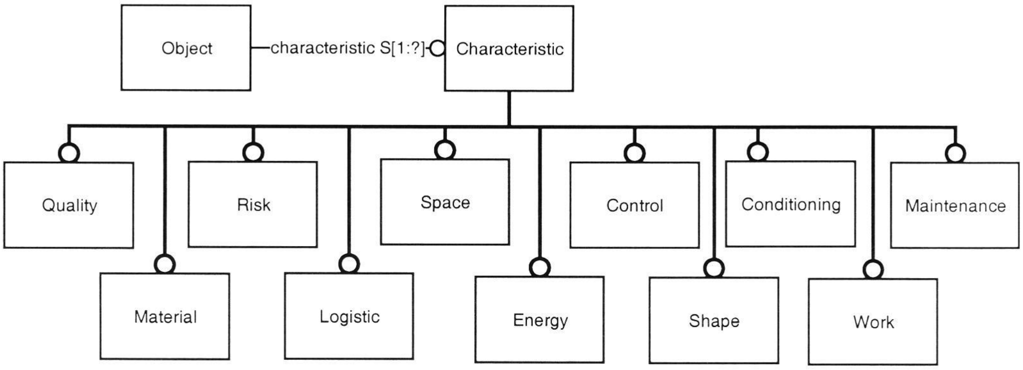


*Figure 7.2 The statement: "Objects are Function Performers" formally defined in Express-G.*

## 7.3.2 *Other Objects are "Related" Objects*

Besides Function Performers the OT usually also contains related Objects that are not Function Performers, but constitute considerable amounts of work, or care. Examples are: existing buildings that have to be demolished, cables, wires and ducts that have to be re-routed, etc. In fact most site objects fall into this class.

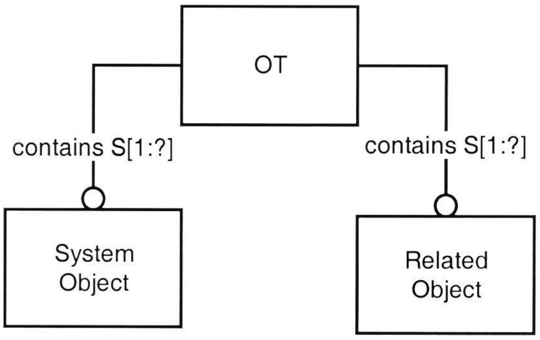


*Figure 7.3. An Object Tree (OT) contains sets of System Objects and sets of Related Objects.*

## 7.3.3 Characteristics

Two types of characteristics are distinguished, common characteristics and private characteristics (private to one discipline, one country, one company, one project, or even to one object).

Most PDT efforts discussed in chapter 5 try to treat all characteristics as common characteristics. This greatly increases the standardization problem. The solution applied in this research is to use only very few common characteristics (only those that are really common to everyone) and to leave the definition of the private characteristics to individual design or construction disciplines.

## 7.3.4 Object Relations

The objects in an OT are related to each other. The two types of object relations used are: (1) decomposition (*contains*), and (2) interface (*interfaces*).

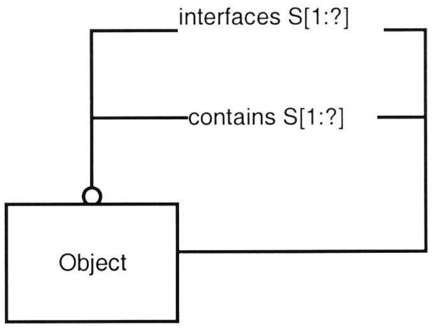


*Figure 7.4 Core of the meta-model of the Object Tree approach. Objects can contain other Objects, and Objects interface with other Objects.*

Interfaces can be Crossings, Connections, Penetrations, etc. Though Express-G does not support specialization of relations, also relationship names should be 'standardized' within the project (and stored in the Taxonomy, see below).

Both decomposition and interfaces will be further discussed later on, in 7.3.7 and 7.5.2 respectively.

## *7.3.5 Name and Identification*

The Object Tree is accompanied by a vocabulary of standardized terms, called ObjectNames stored in a Taxonomy. Each Object has a Name and an Identification. The Name is one of the ObjectNames from the Taxonomy. The Identification is unique for the project. Figure 7.4 below shows the definitions.

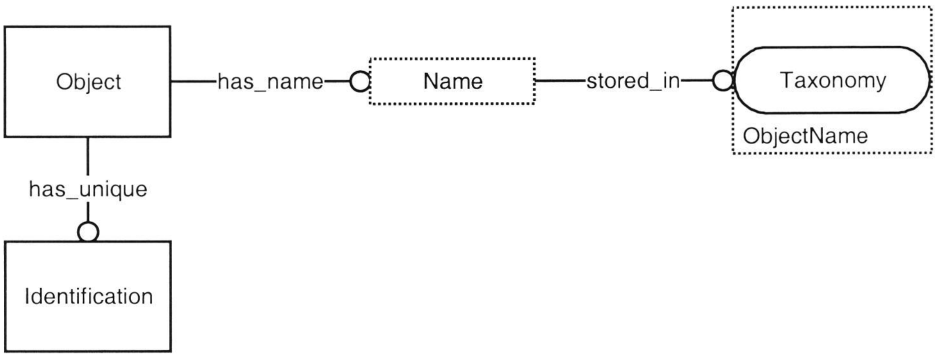


*Figure 7.5. Objects have a name, which is one of the names stored in the taxonomy, and a unique identification.*

The modelling construct for name and identification showed above may lead to objects with a name "Bridge", which is stored in the Taxonomy, and an identification "12345". This object can then be referred to as "Bridge 12345".

The construct could be extended with a free format meaningful addition for communication purposes. This may lead to an object with:

- Name: "Bridge" – taken from the taxonomy,
- Name addition: "on the River Kwai" – a free format addition to the taxonomy name,
- Identification: "12345" – a unique identification code.

For unique reference, such an addition is not needed. But for human communication, such an addition may be very useful (in order to serve as a natural identifier).

The addition mechanism will not be added to the Object Tree meta-model, but could be kept in mind when specific Object Trees are being developed.

## *7.3.6 Function versus Solution*

As discussed in the previous chapters several proposals for modelling function versus solution have been made. A difficulty in modelling functions and solutions is that an Object could be completely specified according to one or more aspects, but also completely unspecified according to some other aspects. A useful solution for this which adequately describes the necessary concepts without making the model too complex, was found in the BCCM [ISO 1994]. Here the same modelling principle is applied, but in a slightly different way.

In the OT approach each object characteristic can have a Status. The Status can for example be 'required', 'proposed', 'realized', 'alternative', 'accepted', 'rejected', etc.

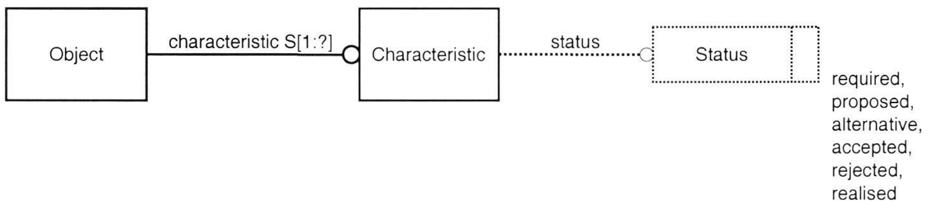


*Figure 7.6 Objects have Characteristics. Characteristics have a Status. The Status expresses the distinction between required, proposed (designed) and realized Objects. Other options are for instance: rejected, alternative, or accepted. Which values for Status are taken into account has to be decided by the project team.*

An important feature of this construct is the possibility to create alternative proposals for a design. This feature has some major advantages in design:

- During the design process, several alternatives can be developed concurrently, and the different options can be kept open until a selection has to be made,
- During operation and maintenance, the Object Tree system can provide information on alternatives that have been considered in the design; this can be very valuable information, for example in case a specific part must be replaced, but can not be replaced by an identical substitute (for example because the supplier does not exist anymore).

The concept of alternative solutions was earlier modelled in the GARM [Gielingh 1988] as part of the Functional Unit/Technical Solution paradigm. See 5.4.2.

In the OT approach, alternative proposals can be created by:

- Creation of an Object with Characteristics that have the Status “proposed”,
- Creation of another Object with Characteristics that have the Status “alternative”.

At any moment in the design process, the decision can be taken to change the status of the Object Characteristics, for example:

- Change from “alternative” to “rejected”,
- Change from “alternative” to “proposed”.

Finally, it is possible in a project that characteristics of an Object always have the same Status. Of course this can be modelled using the construct described above. But in that case it is more convenient to relate the Status directly to Object. This is yet another decision that can be taken by the project team.

## *7.3.7 Objects and Systems*

Each artefact can be divided into a number of smaller Objects. In 2.2.5 a number of decomposition methods have been discussed. Below a few decomposition methods will be recalled and an elaboration is made for the Object Tree.

The first distinction in decomposition methods was the distinction in:

- A requirements-driven decomposition, in which system requirements decompose into subsystem requirements and so on,
- A solution-driven decomposition, in which a system decomposes into e.g. assemblies, subassemblies, components and parts,
- A combination of both.

A combined decomposition method is of course the most complete one: it supports the definition of requirements, solutions and the relationships between them. But for the Object Tree approach, a combined decomposition does not seem the best option. First of all because a combined decomposition method leads to very complex models, as other efforts have shown.

But more importantly, the Object Tree approach does not aim at a complete definition of requirements. Instead, the Object Tree approach focuses on physical objects and their characteristics. For that reason, the Object Tree will make use of solution-driven decomposition methods. In other words: the Object Tree is basically a solution tree.

Next, within the Building and Construction context, three kinds of systems have been distinguished: subsystems, aspect systems and functional systems.

- In subsystems decomposition a system is cut into pieces in a geometric/ geographical way,
- In aspect systems decomposition a system is divided into groups of elements that share a role or aspect,
- In functional systems decomposition a system is divided into groups of elements that together perform a specific function.

One of the conclusions of 2.2.5 was that functional systems, as defined above, often have overlaps with other functional systems, due to the multi-functional nature of many building objects. This implies that functional system decomposition will also lead to a very complex model structure. Therefore, the functional systems decomposition is not regarded as a good candidate for the Object Tree approach either. But again, there is yet another reason for that: in our definition, functional systems have a performance that can be evaluated (calculated, simulated etc.). And again, this is clearly not an objective of the Object Tree approach.

This leaves us with the two remaining decomposition methods: subsystems decomposition and aspect systems decomposition. The Object Tree approach will support both methods.

Figure 7.7 shows the different systems as used in the Object Tree approach. Please note that the grey area with functional systems and performance are considered as out of scope.

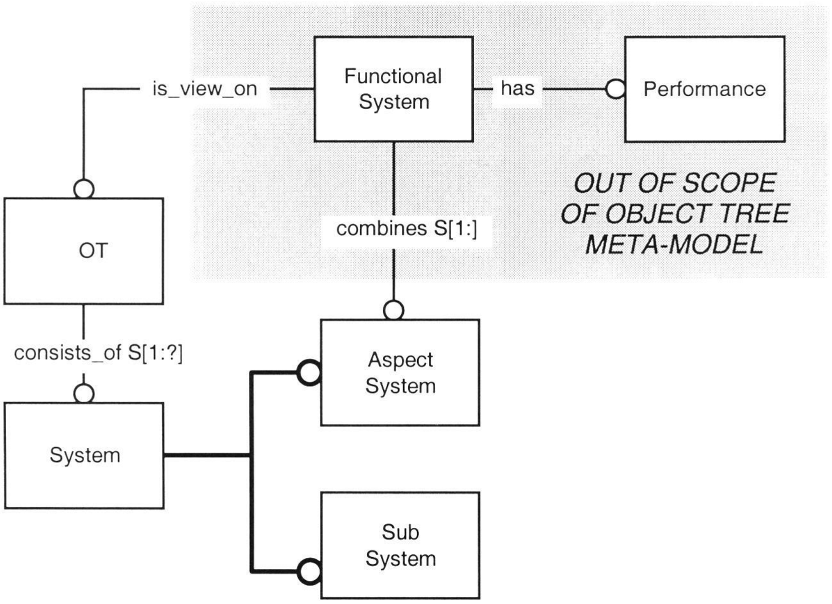


*Figure 7.7. The different systems as used in the Object Tree approach. Please note that functional systems and performance are left out of scope.*

In several existing approaches concepts similar to the subsystems and aspect systems have been used. But it was often also tried to solve the problem: how to relate the Objects in the two decompositions. At first one might think that the two decompositions are orthogonal. For example, a bone in a leg can be found by subsystem decomposition of the skeleton system, or by aspect decomposition of the leg. In general however this idea of orthogonality does not work that way. Often nerves, muscles and blood vessels are not in the least concerned by the shape of our body.

Finally, the reason for drawing the boundary of the OT around the subsystems and aspect systems can be summarized as follows. There are many functional views and functional systems that may play a role in somebody's mind. Supporting all views is simply not possible. It was therefore decided to leave all functional views to the users of the OT and restrict the scope to the other two system types.

### 7.3.8 *Objects Have Owners*

Ownership of Objects (responsibility) is very important for Project Management. Two types of ownership are distinguished. ownership of System Objects and ownership of Interface Objects. System Objects usually have one Owner and Interface Objects have two or more Owners, i.e. those participants that are responsible for the interface.

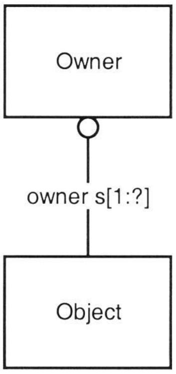


*Figure 7.8 Objects have Owners.*

In the ownership construct, shown in Figure 7.8, the distinction between System Objects and Interface Objects is not made explicit. But both types of objects are covered by the cardinality (an Object has one or more Owners).

## 7.4 The Object Tree Meta Model

Combining the modelling constructs from the previous sections results in the OT meta-model, as shown in Figure 7.9.

Object is the root of the model. It is now modelled as an abstract supertype, because Object itself will not be instantiated. Only the subtypes of Object: System, SystemObject and RelatedObject will be instantiated.

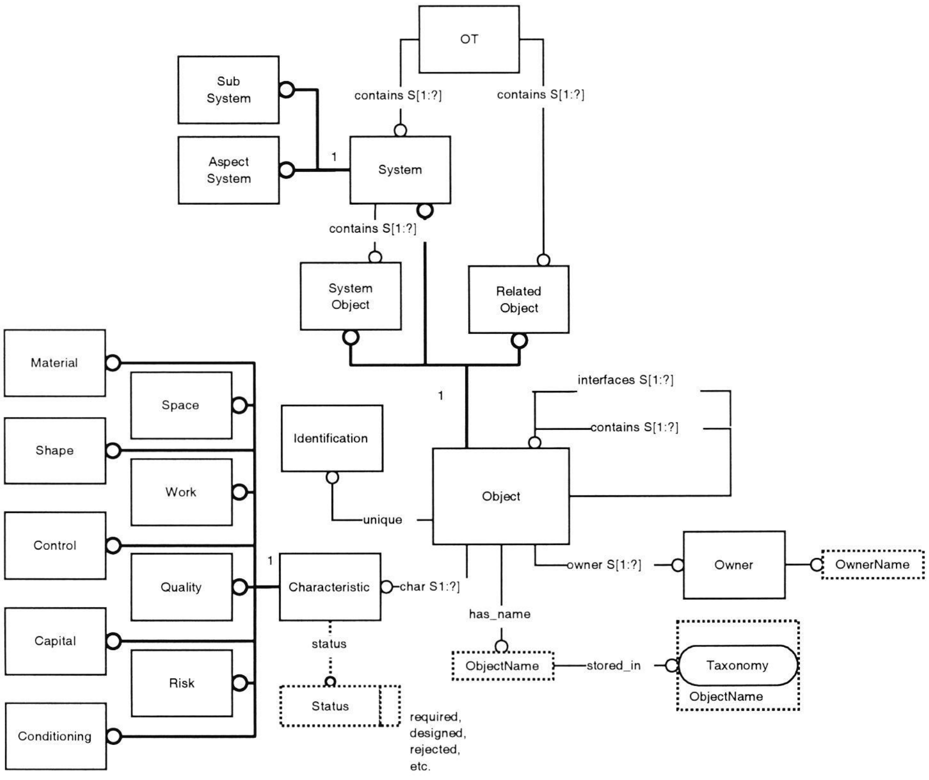


*Figure 7.9. The OT meta-model.*

The meta-model of figure 7.9 will now be extended with attributes that are directly related to characteristics and properties.

## 7.5 Elaboration of Characteristics

In the following sections some important characteristics will be elaborated a little bit further.

### *7.5.1 Shape*

Though shape is not overly important in the OT approach, it must be dealt with. There are several ways to handle shape information. For the OT approach, the chosen method is to add a reference to a (set of) technical drawing(s) that describe the Object in more detail.

This is a very simple approach that can be used both for electronic drawings and paper drawings. Ideally the project partners use one common EDM-system, and the references in the OT should refer to electronic documents contained in the EDM-system.

The combination of an OT and an EDM-system is very powerful and pragmatic. It forms a bridge between the paper based IS of the past and the electronic IS of the future.

Another advantage of this combination is that (1) drawings contain more information than shape, e.g. material, and (2) also other electronic documents, like budgets and schedules can be handled in the same way.

### *7.5.2 Interfaces*

As described above, objects of the Object Tree can have three types of relationships with other objects: 'is_a', 'contains' and 'interfaces'. In the design and construction processes, the interface relationship is extremely important. In fact, interfaces can be defined as areas where things may go wrong.

For that reason, it is not uncommon in complex projects to practice *interface management,* see for example 8.3.2.

In short, such an approach consists of:

- Identification and analysis of (critical) interfaces, for example using interface matrices,
- Identification of interface owners, or interface managers, that are in charge of a proper solution for the interface,
- Support of the design and development of solutions for the interfaces,
- Management of the interface development (using interface progress reports etc.).

Object Trees can play an important role in the support of interface management. Interface management functions that can be supported by Object Trees are:

- The identification of interfaces – a list of interfaces could be obtained by a query on the Object Tree.
- The identification of interface owners, using the ownership construct in the Object Tree.

- The storage of solutions for interfaces, including alternative solutions etc. using the object-status construct.

The actual design of interfaces and the management of interfaces are likely to remain functions that will mainly require human activities.

## 7.6 Implementing Object Trees

Implementing an Object Tree can be done in several ways. As discussed in chapter 4, a number of integration technologies are being developed that can be used for the implementation of Object Trees.

The first and oldest technology discussed in chapter 4 is the work on STEP. It was concluded that, despite the huge efforts spent, STEP has not succeeded to deliver a simple basis for implementation for building and construction.

Secondly there is the work on OMG/CORBA and the latest Internet developments. These technologies are more up-to-date and promise to become the key technologies for electronic communication in building and construction in the near future. But at the time of this research, these technologies were not mature enough to apply in a simple way.

On the other end, there is a range of commercially available software, ranging from low-cost general purpose office applications (Excel, Access) and the latest operating systems and Internet tools to product data management (PDM) systems and advanced object oriented toolkits.

In the remains of this chapter, these four types of software will be discussed.

### *7.6.1 General Purpose (Office) Software*

The first implementation method that comes to mind is to use general purpose office applications such as spreadsheets and databases. Systems such as Microsoft Excel and Microsoft Access are already available in most offices. Therefore the use of such systems does not require any expenses on software. Another advantage is that many users already know how these systems work. Often they have these systems at home too, which stimulates the use and modification of an implementation both at work and at home.

A general disadvantage of such systems is the lack of built-in support of both common functions needed to work with an Object Tree (e.g. filtering, sorting, version management, etc.) and of advanced functions (e.g. links with other systems such as CAD). Also there is a danger that the user developed modifications of the system lead to a badly structured and documented system, which turns out to be very difficult to maintain.

Furthermore, the programs Excel and Access have their specific advantages and disadvantages. Excel has a strong user interface, and it is easy to create an Object Tree in Excel that a large group of users will find easy to use. But Excel is not a database package, and things like database maintenance, data integrity etc. must largely be done by hand. Of course this is quite an awesome job, and it will go wrong sooner or later when the Object Tree gets a considerable size (e.g. hundreds of objects or more).

Access is indeed a database system, so things like database maintenance and data integrity can be taken care of. But a sufficient user interface in Access is harder to get: an Access specialist must define forms, queries and reports. Furthermore, dedicated functions for creating and editing objects of the Object Tree, editing object data and linking objects to e.g. CAD drawings are either difficult to create or just impossible to obtain.

All in all, systems such as Excel and Access do have a low threshold in the beginning, but are only sufficient to support the very basic functions that are needed for Object Trees.

Another possibility is to use Microsoft's Explorer. Particularly the latest versions of the Explorer that support browsing over the Web are interesting for BC. An OT implemented as a hierarchy of files and made available through the standard browsing facility can be readily accessed by 99% of all the participants in a project that share an Intranet (or Extranet). However, with respect to version management and maintenance, Microsoft's Explorer has the same lack of support as Excel.

## *7.6.2 Advanced Operating Systems and Internet Tools*

A second implementation method is to use advanced operating systems and/or Internet tools like browsers. Subsequent operating systems are showing more powerful functionality in each new release. Functions such as sorting and filtering, exploding and imploding lists, search engines, navigation tools, etc. are becoming more and more standard functions.

At the same time, Internet technology is also improving at high speed as discussed in chapter 4. In fact, it will not take long until operating systems and web browsers have merged into a single navigation system on everyone's desktop. At the moment, the introduction of XML is a very important development. XML combined with Java in a powerful tool will eventually provide the necessary means for the implementation of Object Trees on an Internet technology based platform.

At the time of writing, the current operating systems and Internet tools are not yet able to compete with Excel or Access, but they are not far behind. And it may be expected that in the near future (a few years at most), operating systems are superior to applications with respect to functions that are required for Object Trees.

### *7.6.3 Product Data Management Systems*

Probably the oldest commercial systems that are devoted to management of object data, are so-called Product Data Management systems or PDM systems. A PDM system is a system in which product data is managed.

Normally a PDM system at least:

- Has (or controls) a database with product data (products and product properties, such as name and ID) structured in some kind of product tree, and CAD data (drawings and drawing properties, such as name, version, status, owner),
- Has some kind of interface with one or several CAD systems and text editors,
- Provides means for status and version management, usually including a work flow mechanism (for a control and verification procedure of drawings).

Current PDM systems show a variety of extra functions [EDM/PDM Systems 1998], such as:

- Workflow functionality,
- Configuration management functions (e.g. Eigner)
- Integration with EDM/document management,
- Integration with Enterprise Resource Planning (ERP) (e.g. Baan-PDM, SAP),
- Visualization (e.g. Optegra),
- Integration with Internet technology,
- Integration with STEP (e.g. EXPRESS Data Modeller: EDM) and CORBA/OMG (e.g. Sherpa, Metaphase II).

Generally spoken, PDM systems provide both sufficient database functionality and user friendliness. As a result, any PDM system is superior to Excel, Access and to standard operating systems as discussed in 7.6.2.

The most important drawback of PDM-systems is the usually high investment that is needed to buy and implement[10] a commercial PDM system. An investment of many thousands of dollars for every single CAD operator is an investment that is not often found feasible in Building and Construction (this might be different in industries with high-end CAD-stations that are worth more than 100 K$, e.g. the automotive industry).

On the other hand, the necessary investment should not be considered on its own; one should also take into account what efforts are needed by end users and application managers, for example for training and system modifications. Usually, a PDM system requires less training and less modifications, for example because a PDM system has built-in functions that one has to develop on its own if a PDM system is not available. This may very well make a PDM system more cost effective than a solution in Excel or Access.

Another drawback of a PDM system is sometimes the lack of openness and modularity. E.g.

- Some PDM-systems can only work with one CAD system, or have to be modified to work with more than one system,
- Some PDM-systems do support the exchange of documents and drawings, but do not support the exchange of *product data* (the document metadata), for example in conformance with STEP.

When such limitations are of key importance, then an alternative can be found in one of the more advanced object oriented toolkits, for example software developed in STEP-related projects, as discussed below.

### *7.6.4 Advanced Object-Oriented Toolkits*

If neither general software such as Excel or Access, nor specialized software such as PDM-systems are found sufficient for the requirements on the support of Object Trees, then an alternative might be the use of an advanced object oriented toolkit, for

[10] The acquisition of a PDM-system is not just a matter of buying. As with Enterprise Resource Planning (ERP) systems such as SAP and Baan, acquisition means also a huge and expensive effort in tailoring, testing, and education.

example one of the systems that is developed in one of the STEP-related research projects that have been carried out over the years.

Also this group of software systems are available in many flavours. The emphasis of such systems can be on such diverse areas as:

- support of exchange standards such as STEP,
- support of AI/KBE-functionality, e.g. case based reasoning,
- support of true object-oriented mechanisms, e.g. based on CORBA,
- support of the requirements management, i.e. support of the relationship between as-designed product data and as-required product specifications.

Some examples are:

- the EPM system, an object oriented STEP-based toolkit with PDM-functionality that uses EXPRESS for the specification of the database,
- the requirements management system RDD, that supports a well structured specification of product requirements, with possibilities to include the relationship with as-designed product data.

Systems such as these generally do not have the standard functionality of PDM systems. Therefore they require more work to modify the system in order to support the Object Tree requirements, including testing, debugging, documentation etc. Also for this work. Software engineers that are specialized in the toolkit at hand or the technology (e.g. Product Data Technology, object oriented programming, AI) might be needed.

## 7.7 Conclusions

### *7.7.1 The Object Tree Approach*

In this chapter the Object Tree approach has been presented. The main elements of the approach are:

1. The development of the Object Tree takes place on the instance level, i.e. start to model the instances, and worry about object classes later.

2. Objects are considered (1) as physical things that perform a function, or (2) as "related" objects that belong to the relevant context of the primary objects (e.g. site objects).
3. The description of relationships between objects is strictly limited to (1) decomposition relationships (subsystem decomposition and/or aspect system decomposition), and (2) interface relationships.
4. Object Trees can be developed and implemented independent of software applications, vendors, building participants and standardization efforts; in short Object Trees are "neutral".
5. Shape description is delegated to CAD-systems.

### *7.7.2 Implementation of Object Trees*

Because of the simplicity and the neutrality of the approach, quite a few alternatives are available for implementation. Even general office software such as Excel or Access can be used, but such systems offer little support for system maintenance and user-friendliness. A better option is one of the many commercially available PDM-systems.

But in the near future the most promising direction for implementation of Object Trees is in the Internet area. Especially with new developments such as XML, it might become relatively easy to develop powerful Object Tree implementations that are feasible for building and construction practice.

8

# Case: the HSL Object Tree

*This chapter describes a case study where the Object Tree approach has been (and still is being) applied, and evaluates its results. The project is the design and construction of a high-speed railroad line.*

## 8.1 Introduction

This chapter describes experiences with the development and implementation of the HSL Object Tree. The HSL Object Tree is a tool for integration of discipline information in a “real-life” design and construction project: the Dutch high-speed line project (HSL).

### *8.1.1 Context: the High Speed Line project (HSL)*

The High-Speed Line (HSL) project has a simple but huge task: the design and realization of the Dutch part of the new railway track for high-speed trains between Amsterdam and Paris. This is a new track of about 120 kilometres in a densely populated area.

During the case study (1997-1999), the project went from global design to detail design. Recently, preparations have been made for the construction contracts. The first contract has been signed in December 1999, other contracts are expected to follow in the first half of 2000.

The design[11] of the project is carried out by a dedicated project organization of around 500 people. The organization is set up as a single company, with a number of "project offices", each focusing on a part of the engineering task.

In the project organization, many disciplines are represented: civil engineers, railway specialists, ecologists, architects, acoustical specialists, urban planners, etc. Apart from that, the organization includes cost calculators, time managers, etc. These disciplines describe and manage design information in different ways that suit their own needs. At the HSL-project this is clearly shown by the multitude of (object) lists used by the different disciplines. Among others the following lists are used: lists of acoustic screens, of bridges, of areas of land that must be obtained, of construction work packages, cost items, etc., of course all with their own object names, decomposition structure, numbering system, etc.

All of these disciplines have to work together, communicate with each other, and exchange information of the railway to be designed, in order to achieve the project goal. This communication is of course complicated by the multitude of discipline-specific object lists.

In order to enhance communication and information exchange between disciplines, to eliminate redundancies and inconsistencies, and thus to improve the management of design data, an *HSL Object Tree* has been developed.

## 8.2 Goal, Scope, Approach

### *8.2.1 Goal*

The HSL Object Tree is aimed to be the "central list" of objects to be designed. All design disciplines are supposed to use the Object Tree, and to establish an explicit relationship between the HSL Object Tree and their own lists. The objective behind this initiative was the need for improved management of design data. On occasion, the HSL Object Tree has been presented as the "backbone" for the design management.

The HSL Object Tree describes all objects to be designed. This includes all physical objects that will be tangible when the design is realized. It does not include documents

[11] Or more precisely: the preparation of the design-construct agreements, since contractors are expected to do part of the design work.

that describe the product, such as drawings, planning documents, reports etc. It also excludes functional descriptions (e.g. functional systems) or requirements; basically the HSL Object Tree is a *solution tree*.

The product requirements are stored and managed in another system, the so-called *HSL specification tree.* This has been done using the requirements management system RDD. There have been plans to link the requirements tree and the Object Tree, but until now actual links have not been made.

In the HSL project, the following objectives were found very important:

- Integration of existing data structures (specification tree, document tree, work breakdown structure, planning structure, cost tree, etc.),
- Project management tool (e.g. by working with progress control of objects)

Somewhat less important objectives were:

- Checklist for various applications.
- Standardization of object and attribute names (object identification).

### *8.2.2 Scope*

The HSL Object Tree describes:

- All physical objects that are created or modified by the project (including temporary objects and objects to be demolished), with a name and a unique identification,
- A selection of general object properties, such as location, discipline-specific codes and references to documents,
- Decomposition relationships between objects (see later),
- Interfaces between objects.

The HSL Object Tree does not describe:

- "Functional" objects,
- Documents,
- Activities,
- Physical properties (geometry, material etc.); this kind of information can be found on the CAD-drawings, of which references are included in the Object Tree.

### 8.2.3 Approach

The HSL Object Tree has been developed using a bottom up approach. The developers started to gather objects from various disciplines, or rather from various *lists* that circulated in the project organization, e.g. a list of Construction Works, of acoustic screens, of objects to be demolished, of cables and ducts, etc. These objects were put in one big flat list. That was the beginning of the Object Tree.

As with any list, after some tens or hundredth of items the need emerged to create a hierarchical structure. This led to a decomposition structure that is discussed later on.

#### *Differences with Other PDT-Research Approaches*

In contrast to many PDT-research projects, no *type model* has been developed. The HSL Object Tree is, in PDT-terminology, a pure *product model*, i.e. a model of an instance of a product. The reason for this is that the HSL-organization is a temporary project organization that must realize one product (although it is a rather big one).

Another difference with PDT-research projects is that the HSL Object Tree does not define the shape of objects; instead it refers to CAD-drawings. As a result, the HSL-Object Tree does not serve as an "open standard for exchange of CAD/geometry data", that the STEP-related efforts aim at. But that was not the ambition of the HSL-Object Tree. And of course the reduced emphasis on geometry greatly simplifies the implementation of the HSL Object Tree.

## 8.3 Elaboration

### 8.3.1 Objects and Properties

The HSL Object Tree basically consists of *objects* and *object properties*. For each object, the following data are defined: object name, object identification, location, and a selection of other object properties.

#### *Object Names*

For every physical object a common name has been used that is recognized by the project participants. Only a few guidelines are followed with regard to object names:

- object names must be *nouns*. No plural forms such as "buildings to be demolished on the XYZ road".
- object names must be in *singular form.* No activities such as "ground work".
- object names are *preferably unique.* But this is not a hard rule; the unique identification is provided by the object identification.

### *Object Identification*

Every object has a unique identification. It was decided to use *meaningless identifications.* This contrasts with many codings used by design disciplines, in which often subcodes for location, or object type are used.

In fact, the coding of objects started at number 10001 and continued upwards, without even attempting to use sequential numbers for related objects.

This approach led to some resistance from users, but turned out to be more durable than meaningful coding approaches, that sometimes must be changed because of design changes or organizational changes.

### *Location*

In the HSL project, location is a very important property with respect to object searches. As usual in infra-structural projects, the location is (primarily) expressed in a linear way, i.e. by one figure that defines the distance from a project origin along the axis of the track, expressed in km. In short the *kilometrage*. For the HSL, the origin was chosen in the middle of the track, in Rotterdam. Locations north and south of Rotterdam are distinguished by adding a N (for north) or a Z (for "zuid", south), respectively, to the location definition. This approach follows the common approach in the project.

Of course, for the exact position of objects, one has to know the precise position of the project origin in world co-ordinates, and the exact definition of the alignment, i.e. the course of the track axis with its horizontal and vertical curves. These are all available, but are not often used in the daily design work.

## *Other Properties*

Other properties that are stored in the Object Tree are:

- the local government areas that objects are located in (important with respect to getting permissions to build),
- project organizational units that are responsible for the objects (and that have changed many times due to project reorganizations),
- discipline specific codes, and
- references to documents, especially CAD-drawings.

As said earlier, the Object Tree does not store physical properties of objects, such as shape, material, etc. These properties can be found on CAD-drawings or design documentation of the object. However, sometimes physical properties are defined implicitly in the name of object, e.g. "acoustic screen h=4m".

## *Object Detail Level*

For the various object types, the lowest detail levels have been set, for example:

- *construction works* are decomposed until the level of single elements such as bridges, viaducts, tunnels etc. This detail level follows the decomposition level of the HSL List of Construction Works ("Kunstwerkenlijst"). Complex works such as connections of highways are decomposed into its distinct viaducts etc., but not further into columns, beams etc.,
- *acoustic facilities* are decomposed until the level of single screens,
- *cables and ducts* are decomposed until the level of so-called UTOs (Technical Construction Solutions). A UTO is a set of cables and ducts that are treated as a single design issue. Often a UTO consists of a few cables and ducts that are put together in order to reduce the number of crossings of the track. The cables and ducts that a UTO consists of, can be found in the local database that is maintained by the cables and ducts team at the HSL.
- *objects to be demolished* are decomposed into the single buildings or construction works that are to be demolished.

The chosen detail level has led to some 2000 objects situated along the 120 km long track.

## 8.3.2 Decomposition

The decomposition[12] structure of the HSL Object Tree is shown in Figure 8.1.

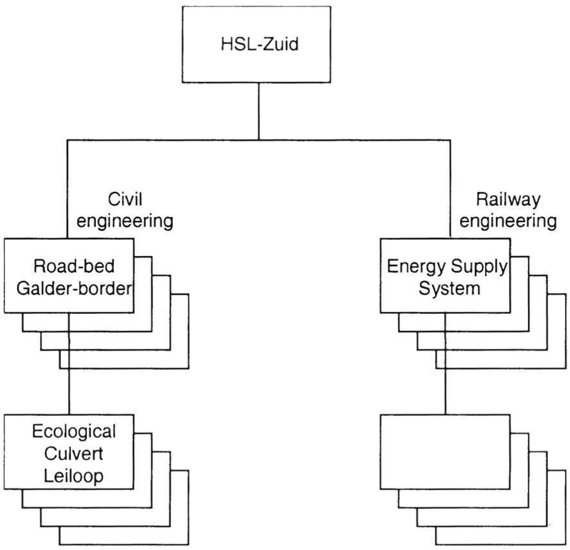


*Figure 8.1 Decomposition structure of the HSL Object Tree*

The top level of the HSL Object Tree is formed by the HSL itself[13].

On the second level a division into two subtrees is made: a civil engineering subtree and a railway engineering subtree. In the two subtrees different decomposition approaches are used, as will be discussed below.

### *The Civil Engineering Subtree*

The civil engineering subtree is decomposed in so-called Track Concepts. A track concept is defined as part of the HSL track with a constant engineering concept that can be recognized in its cross section. For example:

---

[12] Decomposition means "consists of", not "belongs to"

[13] Within the HSL-project also a large amount of work is done on the highways A4 and A16 near the HSL. These highways are also described in the HSL Object Tree. Here they are left out for clarity.

- a track section that is formed by a concrete slab is a track concept,
- a section that is formed by a traditional ground track body is another one,
- a large bridge or tunnel with a specific engineering concept is another one.

This has lead to some 20 Track Concepts, that are shown in Fig 8.2 and 8.3. They vary in length from approximately 1 km to 13 km.

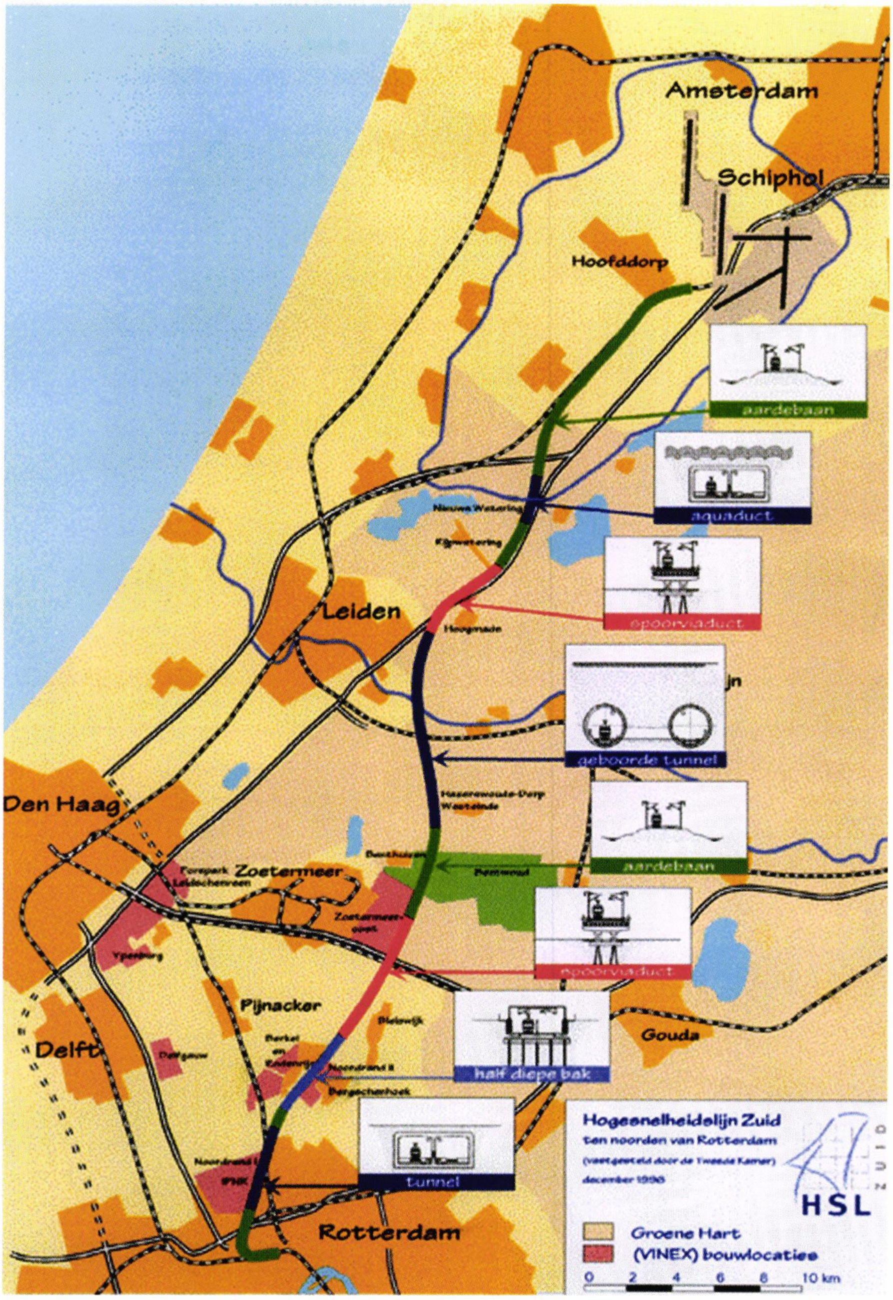


Fig 8.2 HSL Track concepts north of Rotterdam

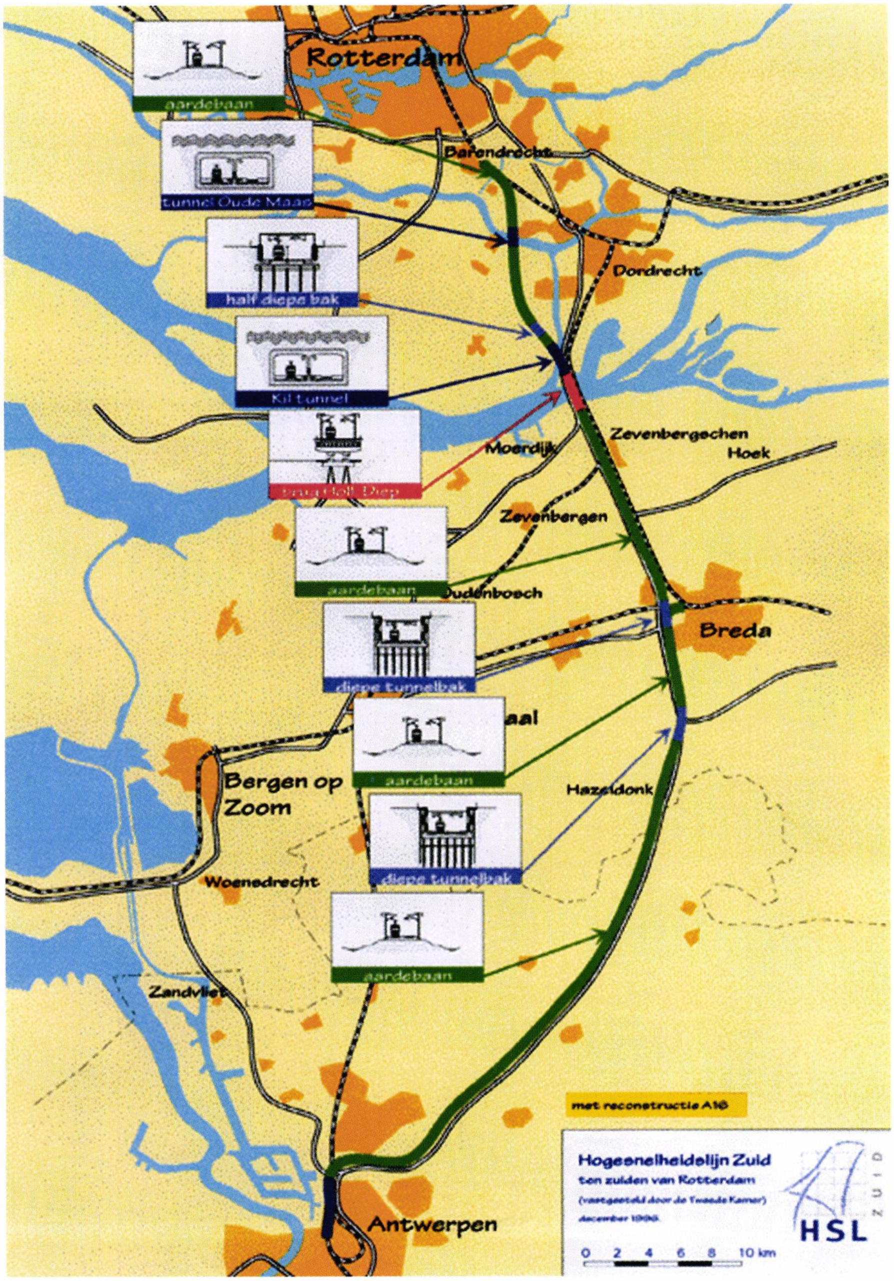


Fig 8.3. HSL Track concepts between Rotterdam and the Belgian border

Note that the decomposition into Track Concepts is used for *all civil* objects, not just the track. That includes all crossing roads and waterways, acoustic screens, cables and ducts. This means that the decomposition in Track Concepts is basically a *geographic decomposition,* since a Track Concept consists of all HSL objects that are located within its boundaries. Or in the terminology used in this thesis: a *subsystems decomposition.*

On the third level the Track Concepts decompose into objects such as construction works (kunstwerken), track sections, acoustic screens, crossed roads, buildings to be demolished, UTO's (Technical Construction Solutions, see 8.3.1). In most cases, level 3 is the lowest decomposition level. But in some cases, level 3 objects decompose further, for example:

- A construction work decomposes into parts that cross a road and a railroad respectively,
- A construction work to be *moved* , decomposes into a work to be *demolished* and a work to be *built.*
- A bridge decomposes into the (concrete) bridge core, and the driveways (ground work) on either side of the bridge.

### The Railway Engineering Subtree

The railway engineering section on level 2 decomposes into railway *systems:* rail/track system, energy supply system, telecommunication system, safety control system, etc. These systems are not bounded geographically, they often stretch along the whole track. In the terms of this thesis, this is an *aspect systems decomposition.*

Next, the systems decompose into single objects on level 3. Which kind of objects depends on the system.

## 8.3.2 Interfaces

Interfaces between the two subtrees are defined both implicitly and explicitly.

### Implicit Interfaces

Implicit interfaces are defined by the object locations: if two objects have the same locations, then it can be assumed that they interface.

Of course this is not a very precise definition of the interfaces. Yet it is used quite often, because it is easy to find objects that are located close to each other, since the use of kilometrage is wide-spread in the project.

## *Explicit Interfaces, Interface Management, Risk Management*

For part of the Object Tree, i.e. the most southern Track Concepts in the civil engineering subtree, (the Brabant section, from Hollandsch Diep to the Belgian border), explicit interfaces have been defined.

The definition of explicit interfaces has been done for two main reasons:

1. To enable interface management

   The idea is that for every interface, an interface manager must be identified, who is in charge of the proper engineering of the interface.

2. To serve as a basis (checklist) for risk analysis and risk management

   This is done with the experience in mind that "interfaces is where things might go wrong".

Another objective behind these goals is the management objective to improve *awareness* of interfaces and risks in the project team.

It is possible to define many types of interfaces between all objects. This could lead to a number of interfaces that is probably no longer manageable. Therefore two arrangements have been made: objects have been clustered and the number of interface types is reduced to four.

The Brabant section consists of some 800 level 3 objects in 8 Track Concepts. This could lead to some 64.000 interfaces per type. Therefore the following level 3 objects have been clustered:

- Wet and dry culverts
- Cables and ducts
- Acoustic facilities
- Green facilities (trees, bushes)

Construction works such as bridges are not clustered, they are treated as single objects. The clustering of objects has led to some 100 objects and clusters. These are divided in four sections, for each of which a relation matrix is set up. In these matrices some 2000 relationships can be defined.

As said above, the number of interface types has been reduced to four. The interface types that are analysed are:

- Geometry, i.e. the shape of object A has a relationship with the shape of object B,
- Deformation, i.e. a deformation of object A affects object B,
- Load relationships, i.e. object A rests on or carries object B,
- "Functional" interfaces, i.e. interfaces related to safety and accessibility.

The first three types are closely related: deformation is proportionate to distance (geometry) per time unit (dx / dt), while load is proportionate to distance (geometry) per time unit raised to a square ($dx / dt^2$). The fourth interface type, the "functional" interface, cannot be related to the other three types.

Another very important interface in construction is the time interface. E.g. the construction of object A must be finished before the construction of object B can start. But since time interfaces at the HSL are managed by the planning system, they are not analysed here.

Based on the above, interface matrices have been made. An example is shown in Fig 8.4.

| | GALDER - GRENS HSL km 50,17-54,54 | 1 A-16 Galder-grens (vanaf 67,5) | 2 Sectiegrens A-16 Galder (67,5) | 3 HSL Aardebaan Galder-grens (> 50,17) | 4 Sectiegrens HSL Galder (Tunnelbak 50,17) | 5 Belgie | 6 Railsystemen | 7 Toe-en afritten A-58 knooppunt Galder | 8 Nw. Ginnekensebaan | 9 Hazeldonksestr. | 10 Ontsluitingswegen Bedrijventerrein H'donk | 11 Duikers | 12 UTO's | 13 Geluidwerende voorzieningen | 14 Belendingen | 15 Inpassingszones | 16 Bereikbaarheid | 17 Veiligheid |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | A-16 Galder-grens (vanaf 67,5) | ▨ | G,V | G,V,B | | G | | G | G,V,B | G,V,B | G,V,B | G,V,B | G,V,B | G,V,B | G | | F | F |
| 2 | Sectiegrens A-16 Galder (67,5) | | ▨ | | | | | | | | | | | | | | | |
| 3 | HSL Aardebaan Galder-grens (> 50,17) | | | ▨ | G,V | G,V | G,V,B | | G,V,B | G,V,B | G,V,B | G,V,B | G,V,B | G,V,B | G,V | | | |
| 4 | Sectiegrens HSL Galder (Tunnelbak 50,17) | | | | ▨ | | G,V | | | | | | | | | | | |
| 5 | Belgie | | | | | ▨ | G,V | | | | | | | | | G | F | F |
| 6 | Railsystemen | | | | | | ▨ | | G,V | G,V | G,V | | | | | | | |
| 7 | Toe-en afritten A-58 knooppunt Galder | | | | | | | ▨ | | | | | | | | | | |
| 8 | Nw. Ginnekensebaan | | | | | | | | ▨ | | | | G,V,B | G,V,B | G | | F | F |
| 9 | Hazeldonksestr. | | | | | | | | | ▨ | | | G,V,B | G,V,B | G | G,V | F | F |
| 10 | Ontsluitingswegen Bedrijventerrein H'donk | | | | | | | | | | ▨ | | G,V,B | G,V,B | G | G,V | F | F |
| 11 | Duikers | | | | | | | | | | | ▨ | G,V,B | G,V,B | G,F | G,F | F | |
| 12 | UTO's | | | | | | | | | | | | ▨ | G,V,B | G,V | G,V | F | |
| 13 | Geluidwerende voorzieningen | | | | | | | | | | | | | ▨ | G,V,F | G,V,F | | |
| 14 | Belendingen | | | | | | | | | | | | | | ▨ | G,F | F | F |
| 15 | Inpassingszones | | | | | | | | | | | | | | | ▨ | | |
| 16 | Bereikbaarheid | | | | | | | | | | | | | | | | ▨ | |
| 17 | Veiligheid | | | | | | | | | | | | | | | | | ▨ |

RAAKVLAKKEN: G=Geometrie; V=Vervormingen; B=Belastingen; F=Functionele relatie

*Fig 8.4 Interface matrix based on the HSL Object Tree, with clustered objects (G=Geometry, V=Deformation, B=Loads, F=Functional).*

A particular objective of the interface analysis and especially the risk analysis is to use these analyses in the tender process of the contractor who will realize the project. The idea is that contractors, who prepare an offer, are requested to do a risk analysis themselves. In the selection process, the risk analyses by the contractors will be included in the assessment of the offers.

Ideally, the contractor will be selected who has the “most economical offer”. This will be the offer with the lowest result of (1) their price and (2) the capitalized risk that the HSL project organization (the client) gets. In order to be able to calculate this capitalized risk, there must be a proper risk analysis by the HSL project organization, and risk analyses by the contractors who make an offer that can be compared with each other. This means also that contractors must receive instructions and guidelines on how to present their risk analysis, including a standard format.

After the selection process, interface and risk *management* must be set up, and these must become a key instrument for project control.

The approach described above is used in the tender process of the HSL project in 1999. Risk analyses have been carried out using the RISMAN-method, an established Dutch method for risk analysis for infrastructure works [RISMAN 1998]. Further elaboration of this subject is beyond the scope of this research.

### *Discussion*

The examples given here show that Object Trees can play a key role in the management of large scale construction projects. A sound explicit definition of objects and their interfaces provides a basis for interface management. And a sound analysis of the key interfaces provides important input for risk analysis and, subsequently, for risk management.

However, as experiences at the HSL-project have learned, a few pitfalls must be avoided. First of all, the implemented Object Tree must be easy to use, and therefore focused on the most important object data only. Too much complexity and too much details are serious dangers here. This is not only because of the "data explosion" when objects are laid out in an interface matrix, where the number of possible interfaces is equal to the square of the number of objects. The main danger is the explosion of necessary meeting hours in a large project, spent on discussion of structure and contents of the Object Tree with a considerable number of ignorant and reluctant colleagues.

Effective support by an information system for Object Trees is another, strongly related prerequisite. This will be discussed further in 8.4.

Another success factor for product data management, interface management and risk management is its integration in the day-to-day project management processes. These subjects should not be delegated to analysts; they should be regarded as prime tasks of the project management team. For example, interfaces and risks should be default agenda items in management team meetings.

This is especially the case in contract types such as Design-Construct, since the success of this kind of contracts depends very much on proper risk management, based on the idea that the participant that is best equipped for the job must manage risks [De Ridder 1994].

The final prerequisite for the success of PDM, interface management and risk management probably lies "between the ears of the managers". One could develop a great system or methodology for PDM, interface management or risk management – when those finally responsible in the project do not use it, all efforts are in vain. This requires what is often called "vision", in the sense of *willingness to improve*, thus to change, to learn new things, to experiment with new developments (e.g. ICT-developments), and to take the risk that new developments eventually do not fulfil the expectations.

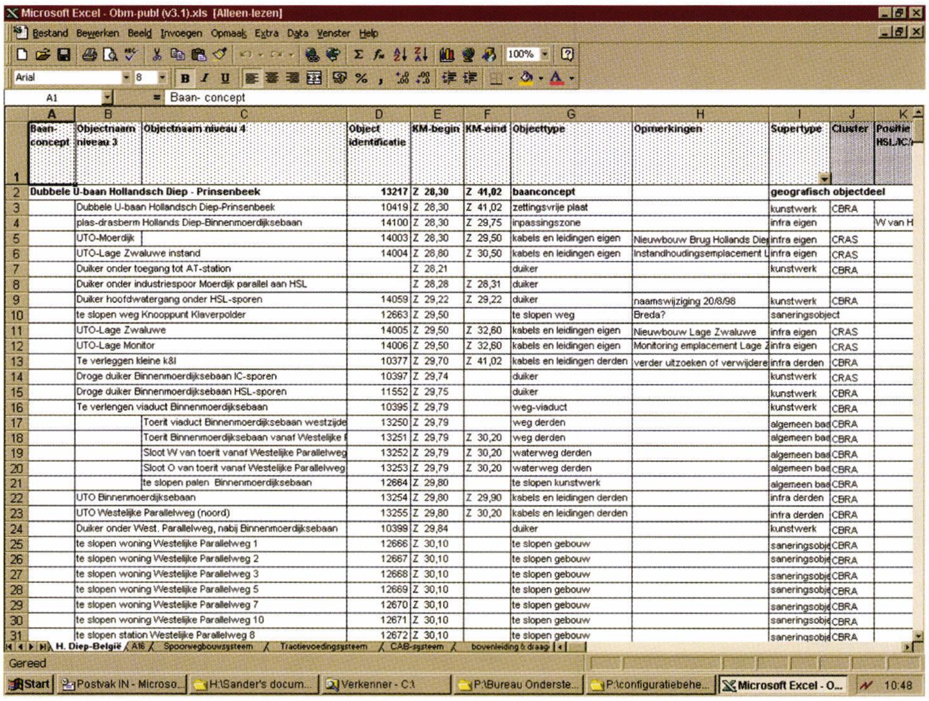
These final remarks are of course not exclusive for the development of Object Trees. They can be said about any innovation. In that sense, this discussion has moved beyond the scope of this research.

## 8.4 Implementation

Over time the HSL Object Tree has been implemented in different ways. Below three main implementations will be discussed: (1) in Excel, (2) in Access and (3) in the PDM-system SmarTeam.

### 8.4.1 Excel

The first implementation of the Object Tree has been in MS Excel. This implementation was basically an Excel file with a list of all objects and its properties. See Fig 8.5.



| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Baan- concept | Objectnaam niveau 3 | Objectnaam niveau 4 | Object identificatie | KM-begin | KM-eind | Objecttype | Opmerkingen | Supertype | Cluster | Positie HSL/IC/ |
| 2 | **Dubbele U-baan Hollandsch Diep - Prinsenbeek** | | | **13217** | **Z 28,30** | **Z 41,02** | **baanconcept** | | **geografisch objectdeel** | | |
| 3 | | Dubbele U-baan Hollandsch Diep-Prinsenbeek | | 10419 | Z 28,30 | Z 41,02 | zettingsvrije plaat | | kunstwerk | CBRA | |
| 4 | | plas-drasberm Hollands Diep-Binnenmoerdijksebaan | | 14100 | Z 28,30 | Z 29,75 | inpassingszone | | infra eigen | | W van H |
| 5 | | UTO-Moerdijk | | 14003 | Z 28,30 | Z 29,50 | kabels en leidingen eigen | Nieuwbouw Brug Hollands Die | infra eigen | CRAS | |
| 6 | | UTO-Lage Zwaluwe instand | | 14004 | Z 28,80 | Z 30,50 | kabels en leidingen eigen | Instandhoudingsemplacement L | infra eigen | CRAS | |
| 7 | | Duiker onder toegang tot AT-station | | | Z 28,21 | | duiker | | kunstwerk | CBRA | |
| 8 | | Duiker onder industriespoor Moerdijk parallel aan HSL | | | Z 28,28 | Z 28,31 | duiker | | | | |
| 9 | | Duiker hoofdwatergang onder HSL-sporen | | 14059 | Z 29,22 | Z 29,22 | duiker | naamswijziging 20/8/98 | kunstwerk | CBRA | |
| 10 | | te slopen weg Knooppunt Klaverpolder | | 12663 | Z 29,50 | | te slopen weg | Breda? | saneringsobject | | |
| 11 | | UTO-Lage Zwaluwe | | 14005 | Z 29,50 | Z 32,60 | kabels en leidingen eigen | Nieuwbouw Lage Zwaluwe | infra eigen | CRAS | |
| 12 | | UTO-Lage Monitor | | 14006 | Z 29,50 | Z 32,60 | kabels en leidingen eigen | Monitoring emplacement Lage Z | infra eigen | CRAS | |
| 13 | | Te verleggen kleine k&l | | 10377 | Z 29,70 | Z 41,02 | kabels en leidingen derden | verder uitzoeken of verwijdere | infra derden | CBRA | |
| 14 | | Droge duiker Binnenmoerdijksebaan IC-sporen | | 10397 | Z 29,74 | | duiker | | kunstwerk | CRAS | |
| 15 | | Droge duiker Binnenmoerdijksebaan HSL-sporen | | 11552 | Z 29,75 | | duiker | | kunstwerk | CBRA | |
| 16 | | Te verlengen viaduct Binnenmoerdijksebaan | | 10395 | Z 29,79 | | weg-viaduct | | kunstwerk | CBRA | |
| 17 | | | Toerit viaduct Binnenmoerdijksebaan westzijde | 13250 | Z 29,79 | | weg derden | | algemeen baa | CBRA | |
| 18 | | | Toerit Binnenmoerdijksebaan vanaf Westelijke | 13251 | Z 29,79 | Z 30,20 | weg derden | | algemeen baa | CBRA | |
| 19 | | | Sloot W van toerit vanaf Westelijke Parallelweg | 13252 | Z 29,79 | Z 30,20 | waterweg derden | | algemeen baa | CBRA | |
| 20 | | | Sloot O van toerit vanaf Westelijke Parallelweg | 13253 | Z 29,79 | Z 30,20 | waterweg derden | | algemeen baa | CBRA | |
| 21 | | | te slopen palen Binnenmoerdijksebaan | 12664 | Z 29,80 | | te slopen kunstwerk | | algemeen baa | CBRA | |
| 22 | | UTO Binnenmoerdijksebaan | | 13254 | Z 29,80 | Z 29,90 | kabels en leidingen derden | | infra derden | CBRA | |
| 23 | | UTO Westelijke Parallelweg (noord) | | 13255 | Z 29,80 | Z 30,20 | kabels en leidingen derden | | infra derden | CBRA | |
| 24 | | Duiker onder West. Parallelweg, nabij Binnenmoerdijksebaan | | 10399 | Z 29,84 | | duiker | | kunstwerk | CBRA | |
| 25 | | te slopen woning Westelijke Parallelweg 1 | | 12666 | Z 30,10 | | te slopen gebouw | | saneringsobje | CBRA | |
| 26 | | te slopen woning Westelijke Parallelweg 2 | | 12667 | Z 30,10 | | te slopen gebouw | | saneringsobje | CBRA | |
| 27 | | te slopen woning Westelijke Parallelweg 3 | | 12668 | Z 30,10 | | te slopen gebouw | | saneringsobje | CBRA | |
| 28 | | te slopen woning Westelijke Parallelweg 5 | | 12669 | Z 30,10 | | te slopen gebouw | | saneringsobje | CBRA | |
| 29 | | te slopen woning Westelijke Parallelweg 7 | | 12670 | Z 30,10 | | te slopen gebouw | | saneringsobje | CBRA | |
| 30 | | te slopen woning Westelijke Parallelweg 10 | | 12671 | Z 30,10 | | te slopen gebouw | | saneringsobje | CBRA | |
| 31 | | te slopen station Westelijke Parallelweg 8 | | 12672 | Z 30,10 | | te slopen gebouw | | saneringsobje | CBRA | |

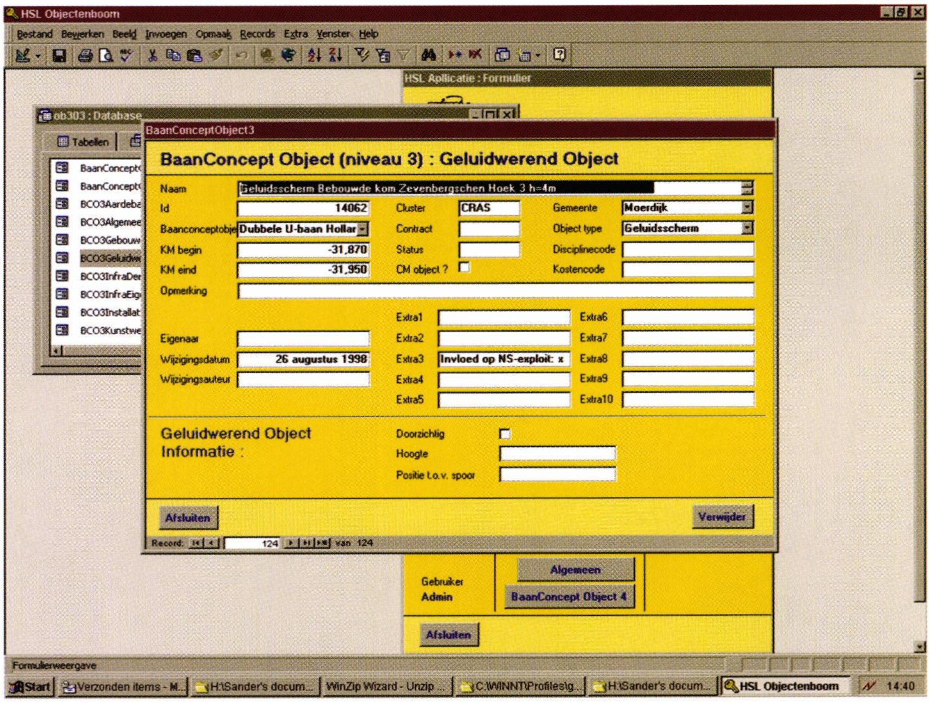
*Fig 8.5 Implementation of the HSL Object Tree in Excel*

The functionality of this implementation was of course limited; no database functions such as queries were supported. The implementation in Excel was done to make a quick start with the HSL Object Tree. The result was an Object Tree that could be used as a complete checklist. The objects were sorted on kilometrage, and kilometrage was the main key for object searches. Some advantages of the Excel implementation were its easy accessibility for a big group of users (since almost everyone has Excel and knows how to use it), and its simple way to present overviews of objects on screen and in print.

## 8.4.2 Access

While the number of objects increased, the maintenance of the HSL Object Tree in Excel became a serious problem. For example, if you would sort the HSL Object Tree in Excel in alphabetical order, it was impossible to go back to the original sequence. Also, it became very difficult to check whether objects were stored more than once. In fact the HSL Object Tree became unmanageable.

For that reason the Object Tree has been converted to an Access database, see Fig 8.6.



BaanConcept Object (niveau 3) : Geluidwerend Object

| | | | | | |
|---|---|---|---|---|---|
| Naam | Geluidsscherm Bebouwde kom Zevenbergschen Hoek 3 h=4m | | | | |
| Id | 14062 | Cluster | CRAS | Gemeente | Moerdijk |
| Baanconceptobje | Dubbele U-baan Hollar | Contract | | Object type | Geluidsscherm |
| KM begin | -31,870 | Status | | Disciplinecode | |
| KM eind | -31,950 | CM object ? | ☐ | Kostencode | |
| Opmerking | | | | | |
| | | Extra1 | | Extra6 | |
| Eigenaar | | Extra2 | | Extra7 | |
| Wijzigingsdatum | 26 augustus 1998 | Extra3 | Invloed op NS-exploit: x | Extra8 | |
| Wijzigingsauteur | | Extra4 | | Extra9 | |
| | | Extra5 | | Extra10 | |

Geluidwerend Object Informatie :

| | |
|---|---|
| Doorzichtig | ☐ |
| Hoogte | |
| Positie t.o.v. spoor | |

Fig 8.6 Implementation of the HSL Object Tree in Access

The implementation in Access meant that the maintainability of the data became a lot better. Double objects became easy to identify, different queries and report functions could be defined etc., all keeping the data in the database consistent. On the other hand, the implementation in Access had two major drawbacks: (1) although it is possible in Access to retrieve all kind of information from the implemented HSL Object Tree, it is necessary to put a lot of effort in query functions etc. to get it all working, and (2) it is also necessary to put a lot of effort in the user interface; for example, the clear presentation of the Excel tables was initially lost in Access.

It is important to note that the kind of functionality that had be developed in Access, is often standard in specialized software, e.g. in Product Data Management software.

It often occurs that there are systems available that offer the needed functionality, but they are not free nor cheap. Instead of buying the more expensive software, design team members sometimes spend a lot of effort in the adaptation of a system such as Access which they already have on their system. In the end they have invested more time and money to get a system which is not as good as what they could have (especially when the development work is done by someone who is not a professional software engineer). This is an interesting phenomenon, but an explanation for this goes beyond our subject.

Back to the Access work. All in all, it was a good decision to move to Access, because of its basic database functionality that enabled a proper database maintenance. An additional advantage was that the Access work provided a sound database that could be used in the next implementation in a PDM-system. It should be noted that most of the Access work turned out to be work on corrections and improvements of the Object Tree data: elimination of double objects, type errors, inconsistencies etc.

### *8.4.3 SmarTeam*

Both the implementation in Excel and the implementation in Access only support the storage and retrieval of data of the Object Tree. But the Object Tree data could be used for many other purposes as well. At least two purposes can be thought of:

- use of the Object Tree for the management of CAD data,
- use of the Object Tree for requirements management.

The first purpose, management of CAD data, has been the main objective of the next implementation of the Object Tree: the implementation in a Product Data Management system. The general objective of the PDM implementation can be described as the integration of the registration, maintenance and exchange of drawings and drawing data, and the registration and maintenance of object data. This is shown in Fig 8.7:

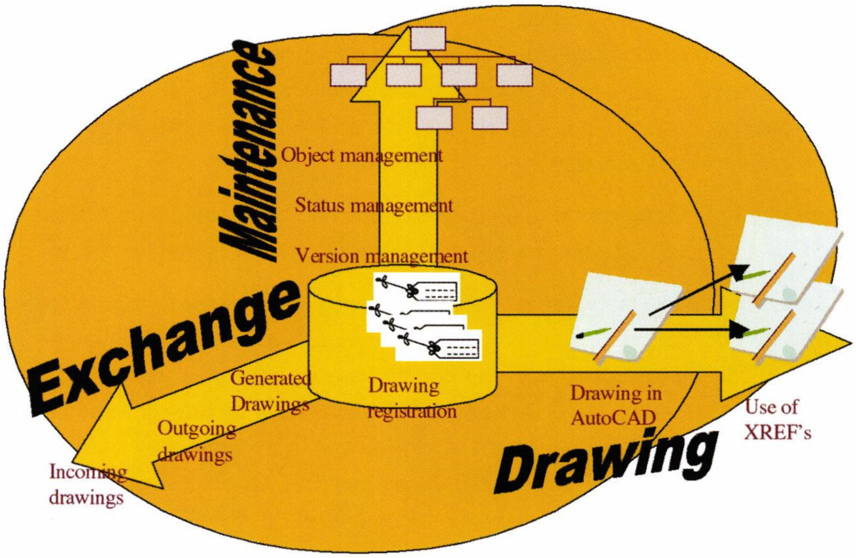
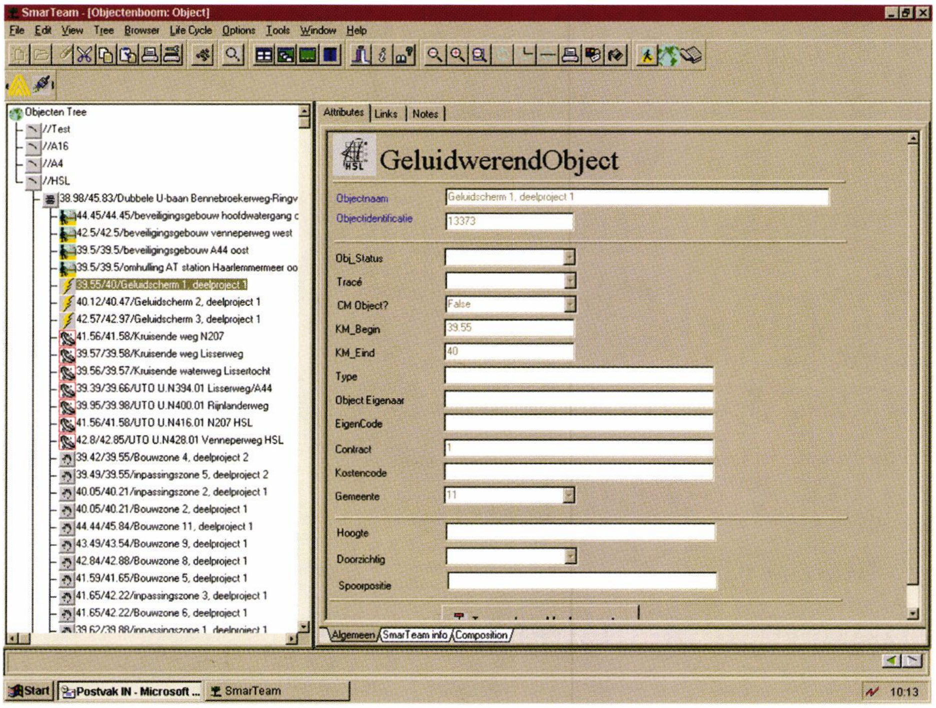


*Fig 8.7 The PDM-implementation of the HSL aims at integration of the Object Tree with CAD production, CAD exchange and management of CAD data (Figure by Dick Laan of MIS/HSL).*

For this purpose, the PDM system SmarTeam has been chosen. In this system two data structures have been stored: (1) the Object Tree and (2) a list (not a tree) of CAD drawings. Both data structures are linked by a reference from the CAD drawing to the object that is represented on the drawing. The reverse relationship, from object to drawing, can be derived by the system. In this way, the user can ask for "the drawings of an object" and for "the objects of a drawing".

Moreover, the user is able to open the drawing in AutoCAD, or a viewer, from within the SmarTeam system. The user can even be forced to do so. In this way, the system can be used to support a systematic procedure for registration and maintenance of CAD data based on the Object Tree.

A screendump of the system is shown in Fig 8.8.


SmarTeam - [Objectenboom: Object]
File Edit View Tree Browser Life Cycle Options Tools Window Help
Attributes Links Notes
Algemeen SmarTeam info Composition
Start Postvak IN - Microsoft ... SmarTeam
10:13

Objecten Tree
- //Test
- //A16
- //A4
- //HSL
  - 38.98/45.83/Dubbele U-baan Bennebroekerweg-Ringv
    - 44.45/44.45/beveiligingsgebouw hoofdwatergang c
    - 42.5/42.5/beveiligingsgebouw venneperweg west
    - 39.5/39.5/beveiligingsgebouw A44 oost
    - 39.5/39.5/omhulling AT station Haarlemmermeer oo
    - 39.55/40/Geluidscherm 1, deelproject 1
    - 40.12/40.47/Geluidscherm 2, deelproject 1
    - 42.57/42.97/Geluidscherm 3, deelproject 1
    - 41.56/41.58/Kruisende weg N207
    - 39.57/39.58/Kruisende weg Lisserweg
    - 39.56/39.57/Kruisende waterweg Lissertocht
    - 39.39/39.66/UTO U.N394.01 Lisserweg/A44
    - 39.95/39.98/UTO U.N400.01 Rijnlanderweg
    - 41.56/41.58/UTO U.N416.01 N207 HSL
    - 42.8/42.85/UTO U.N428.01 Venneperweg HSL
    - 39.42/39.55/Bouwzone 4, deelproject 2
    - 39.49/39.55/inpassingszone 5, deelproject 2
    - 40.05/40.21/inpassingszone 2, deelproject 1
    - 40.05/40.21/Bouwzone 2, deelproject 1
    - 44.44/45.84/Bouwzone 11, deelproject 1
    - 43.49/43.54/Bouwzone 9, deelproject 1
    - 42.84/42.88/Bouwzone 8, deelproject 1
    - 41.59/41.65/Bouwzone 5, deelproject 1
    - 41.65/42.22/inpassingszone 3, deelproject 1
    - 41.65/42.22/Bouwzone 6, deelproject 1
    - 39.62/39.88/inpassingszone 1, deelproject 1

GeluidwerendObject

| | |
|---|---|
| Objectnaam | Geluidscherm 1, deelproject 1 |
| Objectidentificatie | 13373 |
| Obj_Status | |
| Tracé | |
| CM Object? | False |
| KM_Begin | 39.55 |
| KM_Eind | 40 |
| Type | |
| Object Eigenaar | |
| EigenCode | |
| Contract | 1 |
| Kostencode | |
| Gemeente | 11 |
| Hoogte | |
| Doorzichtig | |
| Spoorpositie | |

*Fig 8.8 Implementation of the HSL Object Tree in SmarTeam*

Compared to the implementations in Excel and Access, the SmarTeam implementation provides better data maintenance then Excel, and a better user interface then Access. But the most important advantage is the link with CAD, which stimulates the use of the Object Tree for the management of design data.

## 8.4.4. Network Environment

Also the project situation with offices on different locations, is properly dealt with by the SmarTeam implementation. In Excel and Access, it was necessary to work with multiple copies of the Object Tree file because of network limitations. But the SmarTeam implementation was done using a central Oracle database that can be accessed from different locations. This eliminates inconsistencies between different copies of both the Object Tree and the CAD drawing list. The network structure is shown in Fig 8.9.

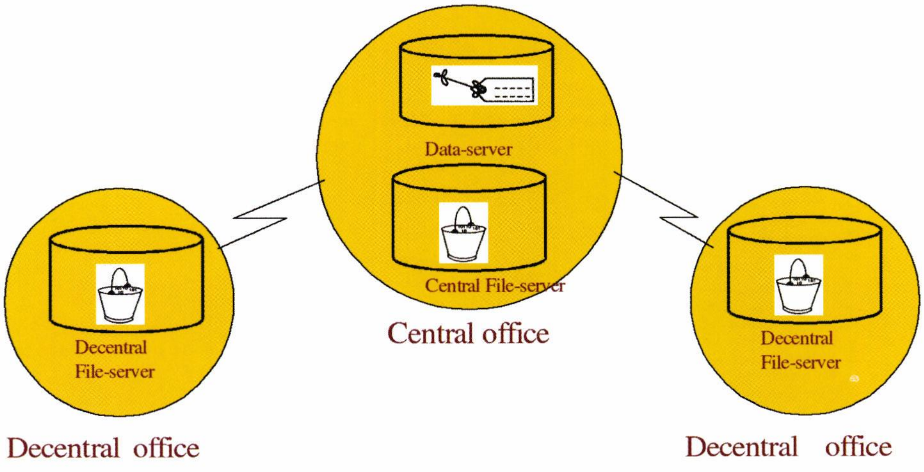


*Figure 8.9 Network structure SmarTeam (Figure by Dick Laan of MIS/HSL)*

In order to reduce data traffic, the central database only stores the object data and the CAD (meta) data (drawing name, date, owner, version, status, etc.), not the CAD-drawings themselves. This implies that for the access of CAD drawings, CAD files do not have to be exchanged between different locations. (Of course, if users on other locations want to open the drawing, then CAD file exchange over the network is indeed needed).

However, in order to get access to the drawings, the central database does have to be accessed for the CAD-metadata. So although only a small amount of data has to be exchanged, the connection with the central database is still necessary.

### *8.4.5 Implementation Alternatives*

As discussed in chapter 7, there are a number of alternative PDM systems on the market, each of which has its specific qualities: PDM systems are specialized in either workflow support, document management support, CAD support, visualization, integration with WWW, ERP, STEP, or support of configuration management (e.g. change control).

For the HSL-project, most of these functions are interesting, but only "nice to have". And because the more interesting systems are quite expensive, they are not a realistic option for the HSL-project. Although the HSL project spends a lot of money on design, a high-end PDM-system was not a realistic option for several reasons: First, the HSL-organization is a temporary organization, where most people only work for a limited amount of time (often less than a year). And second, the HSL Object Tree project started when the design process at the HSL was already halfway.

The most interesting option for the HSL is probably the integration with Internet technology, more specifically with the HSL Intranet, which was already set up in 1997. However, the HSL Intranet is a rather basic implementation of intranet; it offers mostly standard WWW-functions. As a result it would be a rather huge effort to upgrade the current HSL Internet to a useful system for Product Data Management. But for similar future projects, a WWW-based PDM system would certainly be a promising option.

## 8.5 Conclusions and Final Remarks

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 the above: the saying “Keep It Simple & Stupid” is very true in this kind of circumstances: many 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. 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.

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.

Yet the ultimate value of Object Trees depends on such factors as acceptation and management commitment.

# Evaluating the Object Tree Approach

*This chapter evaluates the Object Tree approach and compares its pros and cons with the main stream PDT approach.*

## 9.1 Introduction

In chapter 1 and 2 it was concluded that improved electronic interdisciplinary communication is required to improve the competitiveness of the Building and Construction industry, especially for the execution of large-scale projects. Chapters 3, 4 and 5 evaluated the state of the art of interdisciplinary communication and chapter 7 proposed the Object Tree approach as the best way forward. This chapter evaluates the results of the HSL case study in the light of the industry's needs.

## 9.2 Evaluation

As described in chapter 6 the hypothesis of this thesis is that the Object Tree approach has a number of advantages over other approaches:

- Simplicity and ease of understanding
- Effectiveness

- Bottom up and practical
- Easy to exchange and apply

No advantages go without disadvantages. The following shortcomings have been found.

- Not easily visualized
- Involves a lot of work to develop and maintain

The next two sections discuss to what extend the said advantages are confirmed by the HSL case study.

## 9.3 Pros

### *9.3.1 Simplicity and Understandability*

Simplicity and understandability are very important aspects of a methodology. Especially in the low tech Building and Construction industry advanced and complicated approaches fail due to misunderstanding and reluctance based on ignorance. Everyone who understands the notion of spreadsheets readily understands the Object Tree approach.

The danger of complexity is especially valid for Product Data Technology (PDT). This technology is very important, because it emphasizes the support of semantic exchange of product model data. But most elaborations of PDT have led to too complex model structures.

The Object Tree approach can be regarded as an attempt to achieve semantic exchange of product model data with the simplest possible model structure. This has resulted in a model structure with the following simplifications compared to common PDT-approaches:

- Object Trees are developed as instance models, not as type models[14],
- Objects are physical things that perform functions,

[14] Or in common English: an Object Tree describes objects, not types or classes. Type models can always be developed in a later stage.

- The structure of an Object Tree is a pure decomposition structure, (i.e. a part-of hierarchy, not a type-of or belongs-to hierarchy),
- The decomposition structure is formed by two decomposition methods: a shape-driven subsystems decomposition, and a aspect/role driven aspect systems decomposition,
- Other relationships between objects are limited to physical interfaces only,
- The Object Tree approach is a “neutral” approach: Object Trees can be developed independent of software applications, vendors, building participants or standardization efforts.
- Shape description, i.e. shape definition and representation, is delegated to CAD-systems.

### 9.3.2 Effectiveness

If the Object Tree approach is really effective in large-scale construction projects like HSL is not yet clear. Fact is that the OT approach is being accepted and used. That only is already a victory.

### 9.3.3 Bottom up and Practical

The OT approach is not a long-range top down approach, but suits the project culture of our industry. Participants are used to think ahead until the next project deadline, but little further. Such a project culture requires a bottom up, practical and result-driven approach, as offered by the OT approach.

### 9.3.4 Easy to Exchange and Apply

Interdisciplinary communication can easily be achieved using some simple common tools like a spreadsheet. An OT-based system can support such exchange. For example, a PDM-implementation of the Object Tree as described in 8.4, normally has standard functions to download spreadsheets with object data. In the near future, XML-based tools are expected to provide more powerful means for the exchange of Object Trees.

## 9.4 Cons

### 9.4.1 *Not Easily Visualized*

Visualization of the shape of objects is not the starting point of the Object Tree approach. The idea is to simply include a reference to a technical drawing (first a paper version, later an electronic one). Visualization using VRML or Java3D is however an option that is feasible without too much effort, provided that the electronic parts libraries include VRML or Java3D presentations of their objects.

### 9.4.2 *Involves a Lot of Work to Develop and Maintain*

As long as the type models are not available the development of an unambiguous and complete OT might take quite some time. Also changing the Object Tree might not be as easy as one might think. Adequate support by flexible software may help here, but is only part of the solution.

The development time of the OT can become a serious problem when too much detail and too much complexity is included in the model. Especially in large projects with many different disciplines that must be supported, this may lead to endless meeting sessions spent on discussions of the structure and contents of the Object Tree. In fact this is also a drawback of the instance modelling approach, without a type-model, or in other words, without a database schema.

The key to success here is once again to keep the model structure as simple as possible, and to invest in a good and user-friendly implementation of the Object Tree in e.g. a PDM-system.

## 9.5 Future Extensions

The Object Tree as defined in this thesis is only a first version with a limited scope. The idea is that future extensions should only be considered after the approach has been introduced in practice. However this should not imply that the OT couldn't be used for other applications.

As an example consider the aspect of time planning and control. There is no reason why the OT could not support interdisciplinary communication about start and end dates of design and/or realization processes, with or without a State dimension (required, planned, actual).

As the OT is basically an extensive Object Breakdown Structure (OBS) it seems straightforward to use the OT to derive a Work Breakdown Structure (WBS), thus allocating responsibilities to parties involved in the project.

Another popular application of an OT could be 4D CAD [Aalami and Fischer, 1998]. Can the OT support 4D CAD? Again the answer is positive, but now with a restriction. For 4D CAD the shape representations should be brought in an electronic format, like VRML or Java3D. For simple shapes that is no problem. More complex shapes like curved roads and spaghetti junctions however might require a modeller, for instance using the Shape Deformation Tree representation developed by Willems [1998].

Finally the OT can also be applied for document management and document control. In document control it is important to manage the relationship between Actors (persons and organisations) on the one hand and documents on the other hand. The OT can be used to register which documents are made available to which Actors at what time.

## 9.5 Conclusions

- The Object Tree approach provides a pragmatic, bottom up approach for the development of a simple system for meaningful exchange of product model data in large-scale construction projects.
- As such, Object Trees seem well suited for the support of the short term oriented, low tech building and construction industry. It is not unlikely that only approaches such as the OT approach have a chance of success in building and construction. Many other PDT approaches have failed to provide practical tools for building and construction because they were too much top down and long term oriented.
- Object Trees can also be regarded as a first basis for meaningful exchange of product model data. An important prerequisite for that is the fact that Object Trees are neutral, i.e. independent of software vendors, building participants or standardization efforts. As such, a neutral Object Tree can be extended in a later stage with extra functionality, for example aimed at visualization, time planning and 4D CAD.

10

# Conclusions and Recommendations

*This chapter summarizes the main conclusions of the research and formulates a number of recommendations for the future.*

## 10.1 Introduction

The Building and Construction industry is still too fragmented and conservative to expect ready acceptation and usage of Product Data Technology. The low key Object Tree approach developed in chapter 6 and studied in chapter 7 can be elaborated into an approach that may well serve large-scale building and construction projects. This chapter summarises the conclusions and gives some recommendations for the future.

## 10.2 Recapitulating the Problem

Large-scale Construction projects like the HSL involve a large number of participants and disciplines. Many of them nowadays use computers to support their work. Electronic communication between participants of different disciplines is not possible, at least no meaningful communication (i.e. one that results in understanding).

Consequently the Information System (IS) used in these projects is mainly based on paper (drawings, reports, faxes). The result of all that is:

- That interdisciplinary communication nearly always requires manual information conversion, i.e. information obtained on paper is translated to computer input, and computer output is – again manually – converted to paper based output. In terms of Lean Production [Womack et al, 1991] this is a lot of waste of time and effort, often also error prone that makes the process rigid to change.
- We are now stuck with two ISes, one paper based and another (partial IS) that is electronic. Maintaining consistency is quite a job.
- Both the construction process and the resulting artefacts are not optimal. Either the client's or the taxpayer's money is wasted.

Improving electronic interdisciplinary communication is a worthy cause. How to achieve meaningful electronic communication it is not clear. For over a decade R&D in Product Data Technology provided very little practical tools to the Building-Construction industry. This thesis proposes Object Trees as a means to achieve a significant improvement in interdisciplinary communication in the Building and Construction practice.

## 10.3 Conclusions

In this thesis an approach for Object Trees is developed, and a meta-model is of the object data that is described in an Object Tree is presented. The approach has been implemented and used in the Dutch HSL-project. Based on the experiences of this project, a number of conclusions can been drawn in 9.5. Since these conclusions seem to be not only valid for the HSL, but for the application of Object Trees in general, they are summarized here:

- The Object Tree approach provides a pragmatic, bottom up approach for the development of a simple system for meaningful exchange of product model data in large-scale construction projects.
- As such, Object Trees are well suited for the support of the short term oriented, low tech building and construction industry.
- Object Trees can also be regarded as a first basis for meaningful exchange of product model data. As such, a neutral Object Tree can be extended in a later stage with extra functionality, for example aimed at visualization, time planning and 4D CAD.

## 10.4 Recommendations

Based on the conclusions above, the following recommendations are given.

### *#1 Develop object classifications for different artefacts*

One of the key elements of the Object Tree approach is the internal object classification as described in the Taxonomy. Each type of project (Building, Infrastructure, Plant, etc.) has – at least partly – its own objects. Re-use of object classifications can speed up re-use of earlier work. It is therefore recommended to agree on some kind of national standardization effort in this area. One simple option is to develop electronic libraries of re-usable object classifications. As this is one of the goals of the LexiCon under developed by the STABU, it is recommended that work on the LexiCon will be supported.

### *#2 Develop open electronic libraries of construction products, equipment and processes*

The Object Tree approach also draws on the possibility to reference external electronic libraries of construction materials and components. When also equipment and processes (like Work Methods) are taken into account the same library format and implementation should be available. XML is seen as one of the prime implementation candidates.

### *#3 Develop a number of management tools that build upon the Object Tree*

Implementation of PDT through Design and Engineering has been tried in vain for over a decade. This study concludes that the management role is the best candidate for technology improvements. It is therefore recommended to develop a number of integrated tools, for instance for risk analysis and interface management. These tools will be used to control the project and form the basis of the inter discipline communication required for the future.

### *#4 Perform more case studies*

Re-using the knowledge gained in the HSL-project for other application areas (like complex building projects) is an interesting possibility, which undoubtedly will result in a better understanding of the applicability and limitations of the OT approach.

### *#5 Generalize the HSL Object Tree to a type model*

Once the instance model of the HSL has been generalized and rewritten as an instantiation of the type model, the type model is available for future re-use. Moreover the type model can be augmented with knowledge rules covering various aspects and constraints, as for instance found in the building regulations. As such type models will become valuable assets for the future.

# View Integration in Building Design[15]

## ABSTRACT

The design process of a building is a process of co-operation of many participants, working for different companies. Each participant has his own perception on the building and uses his own building model. Such participant specific building models are called (actor) view models. The combination of multiple view models and multiple companies involved, makes it very difficult to manage communication in the design process.
This paper presents an approach for the management of different view models. Following this approach, discipline specific information is defined in view models. View models can communicate to each other via a model kernel, which is formed by the overlapping of the view models. The view model structure is illustrated with a case study, in which models of a wall are worked out for (1) the structural engineering's view, (2) the energy engineering's view. The integration of these two models illustrates how the kernel model can be constructed. Models for other disciplines can be added to the view model structure. Conceptual design views play a somewhat special role, since it is very difficult to divide conceptual design aspects from one another. Despite of this, two models for conceptual design views are presented and discussed briefly. The presented view model structure can function as a basis for integration of computer applications. Computer programs will be able to communicate to each other through interfaces

---

[15] Paper presented at "Management of Information Technology in Construction", Singapore 1993 [Van Nederveen, 1993]

based on the view models. This seems very well feasible for technical applications such as calculation programs. Regarding CAD systems, results are expected to be more modest, due to the gap between the way designers think, and the way CAD systems work.

Keywords
view integration, building information models, product modelling, integrated design, management of design information

## INTRODUCTION

In the first section the role of views in the building process is described. Next, the notion of views in information technology is discussed and compared to the needs of the building industry.

### Views in Building Design

In a building project, many partners from different companies work together. Every partner uses existing information and creates new information. A lot of the used information is provided by other partners. As a result, a lot of information is exchanged between partners. Fortunately, none of the partners needs all project information. Every partner has a specific task and role in the project. To carry out his task, a partner only needs a subset of the project information. For example, a structural engineer needs information on the structural characteristics of the building and its parts, but he does not need information on thermal characteristics such as u-values, or on visual characteristics such as colour. On the other hand, a partner usually does not process the provided information directly, but transforms it into information he can process. To continue the example, a structural engineer may get shape and material data and transforms this into a structural schema with loads and so on, or into a finite element model. Then he carries out his calculations and provides other partners with the results. To carry out his calculations the engineer creates a specific model of the building which accommodates his specific view. This kind of view specific models are created by every partner throughout the building process. These so called view models allow the partners to work with a considerably smaller amount of information which is appropriate for them to do their job.
An issue which is related to the existence of participants with different views, is the management of responsibility. Partners not only have a specific task in the project, they also have specific responsibilities. The structural engineer is responsible for the structural aspects of the building, the building physicist is responsible for the energetic aspects and so on. As a result of this, partners have to meet agreements on who is responsible for what information, who is allowed to create and edit what information. The distribution of responsibility has a strong influence on the creation of view models. If a partner is responsible for certain aspects, these aspects will be included in his view model. Aspects, which are beyond the responsibility of a partner, will be excluded. In this way, the view mechanism can help to manage responsibility (Van Dam, 1992). Summarized, the view mechanism helps to provide partners with only the information they need, and to manage the responsibility for different design aspects.

## Views in Information Technology

In many recent projects researchers have been working on the conceptual specification of integrated building information models. Well known models in this context are the General AEC Reference Model, or GARM (Gielingh, 1988), the AEC Building Systems Model (Turner, 1989), the RATAS model (Björk, 1989) and the EDM (Eastman, 1991). The aim of these models is to provide an information structure in which all building data can be stored in an integrated way, including information on the building and its parts, and their properties and relations during the different building life cycle stages. This approach is known as product modelling, and is closely related to the development of the ISO standard STEP (STEP, 1992). In all of the mentioned models important abstraction mechanisms, such as decomposition and specialization are worked out extensively.

Following the previous section, the view mechanism seems to be also very important for the management of building information. In database technology, the notion of views is already quite common for many years. This goes back to the ANSI/X3/SPARC reference model for database management systems (Burns, 1985) in which a distinction into internal, conceptual and external views was proposed.

The importance of the notion of views was also recognized in the building information models discussed earlier. In several models, model structures are used which focus on certain aspects of the building. Turner introduced the use of separate models for different building systems, for example the spatial system, the circulation system and the structural system. These systems are described in separate models, each of which focuses on one aspect. A similar approach is used in the RATAS model. The EDM does not define different building systems, but offers mechanisms on a more abstract level. An EDM mechanism which distinguishes different building aspects is the accumulation mechanism. This mechanism focuses on the accumulation of properties of a single aspect in an assembly. In the EDM decomposition is defined as the result of multiple accumulations applicable to the same assembly.

These examples show that the use of aspect specific model structures, is already quite common. However, the discussed models do not explicitly distinguish between discipline specific information. In this paper, this distinction is the starting point, which leads to the formulation of separate discipline view models.

## Overview of Paper

In this paper, the notion of views and view integration is worked out in conjunction with the product modelling approach as used by the models mentioned earlier. In order to do this, the paper starts with an introduction of a number of basic concepts of product modelling and view integration. Next, a case study is presented, in which models of a simple wall are worked out for (1) the structural engineering's view, (2) the energy engineering's view, followed by an integration of both models using a kernel model. After that, the structure of the kernel model is further discussed. Within the kernel so called system models are identified, which describe different building systems.

Then the relationship between views and applications is discussed, while making a distinction into technical applications such as calculation programs, and CAD systems. The paper finalizes with remarks on evaluation and future work.

## BASIC CONCEPTS OF PRODUCT MODELLING AND VIEW INTEGRATION

In this section the basic concepts used for product modelling and view integration in building design are described.

### *Product Model and Product Type Model*

A product model is an information model of a product, in which product data is stored in an integrated way, including information on the product parts, their properties, relations and behaviour, during different product life cycle stages. A product model describes an occurrence of a product, not a class or type. Example: a product model of the Eiffel tower.
A product type model is an information model of a product class, in which general information of the product class is defined in an integrated way, including information on the product parts, their properties, relations and behaviour, during different product life cycle stages. A product type model describes a class of products. Example: a product type model for buildings, or for office buildings.
In the case of real life products, the distinction between product model and product type model is clear: a model which describes an existing building is a product model, a model which describes a class of existing buildings is a product type model.
In the case of product design information, the distinction is not always clear. In most product model literature, and also in this research, it is assumed that a design of a product is described in a product model, in which all design parameters are specified during the design process, including position and orientation. However, the use of such a design model may not be limited to one product occurrence. It can be used for many other products as well (as in the car industry), or as a prototype for a new product model. Thus a single design may serve as a specification for a set of products, which only differ from each other in serial number and in position. In such cases product models can be defined as mere references to a type model, with a position specification (comparable to symbol instances in a CAD system).
On the other hand, existing models such as GARM, the RATAS model and the EDM are clearly not describing product occurrences. Such models describe general information of buildings, or even more general, of AEC products. Therefore, this kind of models are considered as product type models, not product models. The ideas presented in this paper are also dealing with product type model information.

### View Model and View Type Model

A view model describes a discipline specific view on a product. A view model is part of a product model. A view model may overlap with other view models.
A view type model describes a discipline specific view on a product class. A view type model is part of a product type model. A view type model may overlap with other actor view type models.

The goal of these concepts follow from the function of the view mechanism stated earlier: provide partners with only the information they need, and manage the responsibility for different design aspects.
These concepts can be refined by a distinction between disciplines and actors (Tolman, 1993). In this way, the fact that actors may play different roles can be supported. This distinction is absolutely necessary to allow organizational differences in building projects, at least between different countries and ages, but certainly also between projects of almost the same place and time. For example, an architect may play the role of the project manager in one project, but not in another one.

## Kernel Model and Kernel Type Model

A kernel model is formed by the overlapping parts of view models. See figure 1.

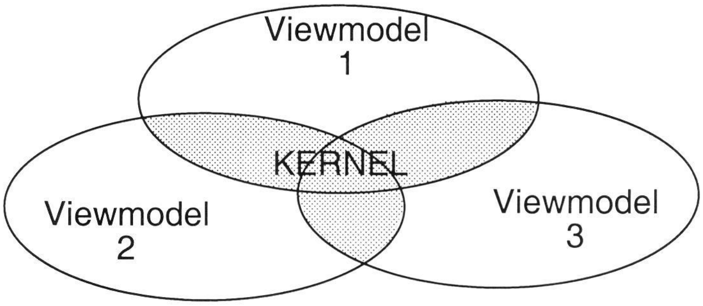


*Fig 1. A kernel model is formed by the overlapping parts of view models. The union of all models is the product model.*

The function of the kernel model is to provide a basis for communication between the different disciplines. As a result of this, the kernel includes exactly all information that is communicated between disciplines.
A kernel type model is formed by the overlapping parts of view type models.

## CASE STUDY: SIMPLE WALL

In this section the concepts introduced in the previous section are illustrated using examples from a case study on a simple wall. The case study fully concentrates on the usage of the view mechanism. Therefore many other aspects of product modelling of buildings will remain out of scope.
These aspects include:

- the decomposition of buildings,
- the relationship between spaces and physical objects,
- the usage of generic concepts which can be applied to all objects, for example modelling constructs defined in STEP resource models.

These aspects will be discussed briefly after the example.

## The structural engineering's view

First, a model is presented which describes the view of the structural engineering discipline on walls: a structural engineering's view type model. Such a model contains the information, which is needed or produced by structural engineers. This includes:

- an idealized shape description,
- an idealized material description,
- load information,
- information on mechanical quantities (moments, forces, elasticity parameters)

  (since decomposition is out of scope, the wall is considered as a single object, which is not decomposed further).

Both the shape description and the material description should be idealized for the structural engineers' needs. This means that these descriptions do not include information which is irrelevant for the structural engineer. For example, a complex shape may be simplified to something like a mechanical schema. In this schema information such as colour data or detail design information, which have no impact on the structural design are left out. As a consequence, the shape and material description in this model is different from descriptions used by other disciplines.

The information discussed here, can be described using the modelling technique NIAM. The symbols in NIAM diagrams must be interpreted as follows:

- circles represent object types,
- double boxes represent relationships,
- arrows represent subtype-relations

For further description of the NIAM modelling language see (Nijssen 1989).

The NIAM diagram for the structural engineer's view on a simple wall may thus look like in figure 2.

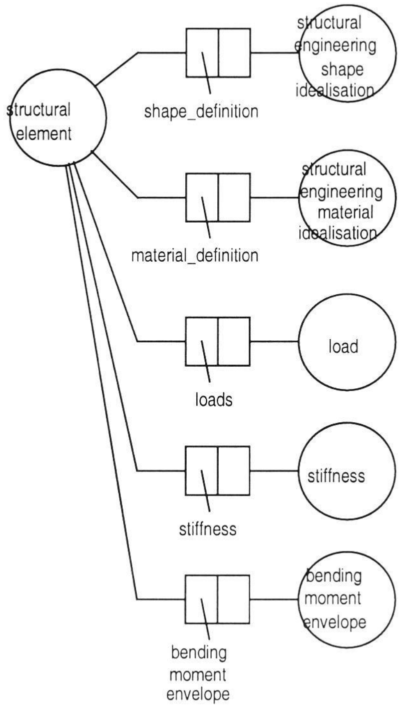


*Fig 2. Structural engineer's view type model of a simple wall.*

## The energy engineering's view

In the same way, a model can be defined for the energy engineering discipline. An energy engineering's view type model contains information, that is needed or produced by energy engineers. This includes:

- an idealized shape description,
- an idealized material description,
- information on energetic quantities, such as transmission coefficients, u-values and so on.

Once again, the shape description and the material description are idealized for the disciplines needs. In this case information such as reinforcement details, are left out, since they are irrelevant for the energy engineer. Obviously the shape and material descriptions will differ from the descriptions in the structural engineer's view type model.

The information discussed here may be modelled in NIAM as follows:

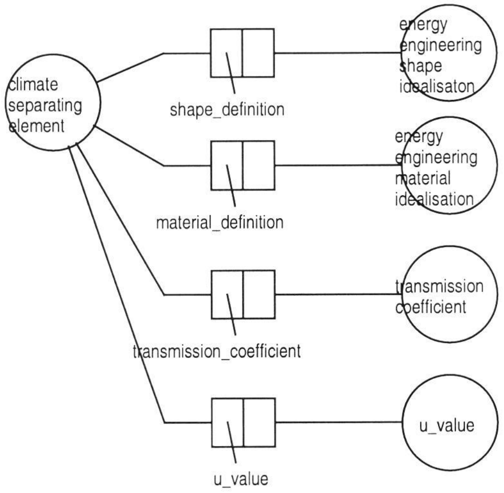

climate separating element
shape_definition
material_definition
transmission_coefficient
u_value
energy engineering shape idealisaton
energy engineering material idealisation
transmission coefficient
u_value

*Fig 3. Energy engineer's view type model of a simple wall.*

## Integration of the view type models

The next step is the integration of the two view type models. Since actors such as the structural engineer and the energy engineer need to co-operate in the design process, it is essential that they can exchange information between their models. This means that the relationship between the models must be specified.
The following diagram shows how this can be done.

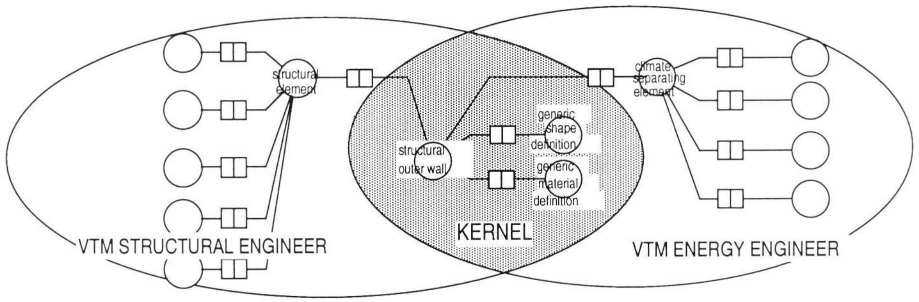


*Fig 4. Integration of the view type models.*

The diagram shows that the view type models are integrated through a general entity 'structural outer wall', which has a 'generic shape definition' and a 'generic material description'. These generic descriptions are used in the communication between disciplines. The view specific descriptions of shape and material, may be derived from this generic description.
Note that the entities 'structural element', 'enclosure element', and 'structural outer wall' are all describing the same object. The information of the outer wall is distributed over the three

entities. Part of this information are three different shape descriptions, which can be derived from one another. The rules for these derivations are included in the relationships between the view entities and the kernel entity. This type of information is view specific, so the relationships are not part of the kernel.

## THE STRUCTURE OF THE KERNEL

In the presented example, the contents of the kernel remains small and manageable. However, when other views are added, such as the architect's view and the HVAC engineer's view, the kernel will become a lot more complex. This is due to the fact that the kernel includes all information that is communicated between participants, which followed from the function of the kernel: to provide a basis for communication between the different disciplines.

### System Models

For this reason it is necessary to add structure to the kernel. For this purpose, concepts of the existing building models discussed earlier, can be used. For instance, different building systems can be described in separate models, as in (Turner, 1989). Following this approach the following systems can be distinguished:

- the functional system,
- the space / enclosure system,
- the structural system,
- several HVAC systems.

The models of these systems can be regarded as aspect models: the models describe a certain aspect of the building, which corresponds to a certain set of properties. However, since the term aspect models is used with other meanings as well, we prefer to use the term system models here.

The difference between system models and view models may need some clarification. A system model is defined by an aspect of the product. It is considered as part of the kernel. It contains only information, which is used in the communication between partners. Reversely, it does not contain information which is only used by one partner. Finally, a system model uses a common language which is understood by all disciplines.

A view model is defined by the view of a discipline. It partly overlaps with other view models. The overlapping parts belong to the kernel, and may belong to system models. The private part of the view model uses a discipline specific language, or jargon. Nevertheless, system models and view models may have a strong relationship. It is clear that the structural engineer's view model will be heavily related to the structural system model.

### The Functional System

Within building information, a subset can be defined which concentrates on the function of the building and its parts and the relationships between these functions. This subset is called the functional system. The functional system is limited to space functions. The reason for this is that physical objects usually have multiple functions (bear load, separate spaces), which are easier to describe in other system models. Thus the functional system can be seen as the subject of functional analysis in a design process.

The functional system model starts with a general specification of the function of the building. This may be something like 'an office housing facility for 200 employees with facilities to have meetings with groups up to 50 people...'. The general building function can be decomposed into space functions. Furthermore, space functions can be decomposed into smaller space functions, but the number of decomposition levels cannot be fixed. These considerations lead to a rather abstract model, with a recursive decomposition construct, see fig 5.

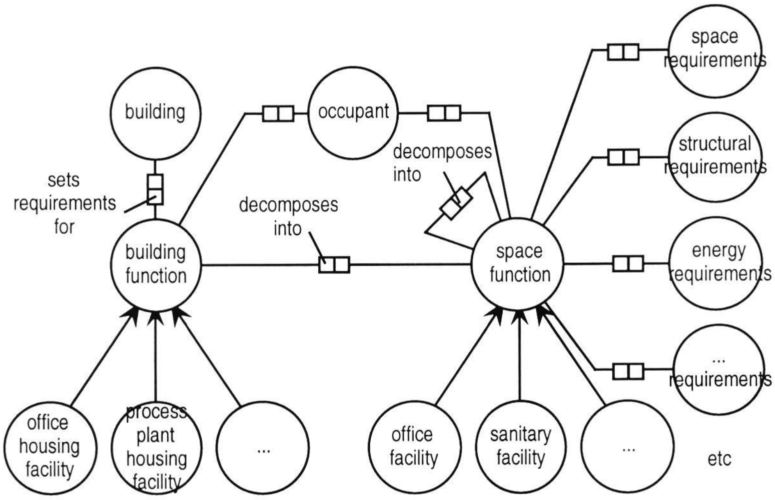


*Fig 5. Overview of the functional system type model.*

One could argue that building function is in fact also a space function, which accidentally is fulfilled by a building. This idea would lead to an even more abstract model, with only one entity space function.

The requirements (on the right side in the NIAM diagram), which are derived from the function of a space, form an interrelation with the other system models, such as the structural system model, and the space / enclosure system model, which is discussed below.

## The Space / Enclosure System

The space / enclosure system of a building is the collection of spaces, space boundaries and enclosing structures of the building. The relationships between these entities have already been worked out in a number of models. Björk (1992) has recently published an overview of four of these models (the RATAS model by VTT, the Integrated Data Model by CSTB, the House Model by De Waard and the Synthesis Model by GSD), and has added a new model in which the other models are synthesized. In this study many important aspects of the space / enclosure system are identified and worked out quite extensively. The synthesized model may not be perfect, but it is seems to be a good starting point for the space / enclosure system model. Figure 6 shows an outline of the synthesized model in NIAM. In this outline the assembly entities are left out, as well as subspaces (spaces which are only partially bounded by physical elements), and component layer information. For the complete model see (Björk 1992).

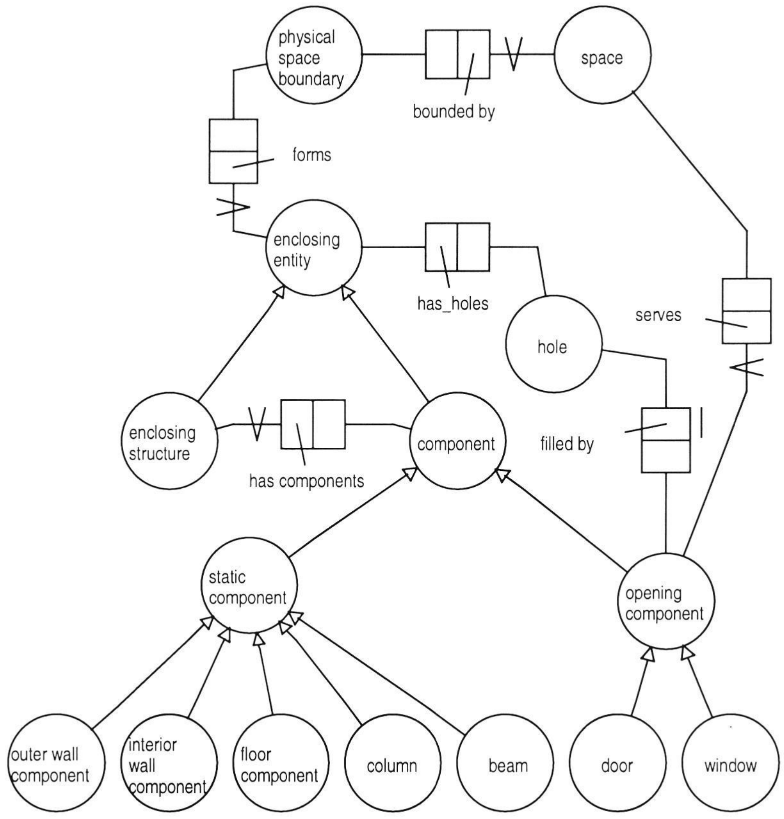


*Fig 6. Outline of the space, space boundary and enclosing structure model by Björk.*

## VIEWS AND APPLICATIONS

One of the intentions of the view model approach is integration of computer applications used in design. For example, information can be interchanged between a structural design application and an energy calculation program through interfaces based on the discussed view models and the kernel model. This approach seems very well feasible for specialized disciplines such as structural engineering, energy engineering, HVAC design and so on. The data structures of the applications used in these disciplines usually fit quite well in the structures of the view type models.

This picture changes if tools used for conceptual design are considered, such as CAD systems. The problem is that the data structure of a CAD system is usually geometry oriented, and contains little notion of functional, spatial and physical properties and structures. In fact, there is a significant gap between the way designers think, and the way CAD systems work. The

ideas presented here can not only be regarded as an approach for meaningful data exchange, but also as a starting point for future design systems, which really work the way designers think.
As long as such systems are not available, we must try to achieve the best possible exchange between the current systems. For this purpose, the view approach is also useful. The models will become more simple, and the exchanged data will not be as meaningful as we think it can be. But progression of the data exchange between current systems is already of great value for the building industry.

## EVALUATION AND FUTURE WORK

In this paper an approach for the integration of different views in building design is presented. For this purpose, view models and view type models are defined, and illustrated with examples. The models are not yet complete, and it is still a lot of work to complete them. Nevertheless, the ideas on view integration already become clear in the presented models.
Other future work concerns implementation efforts. The conceptual models as presented here must be tested in implementations to find out whether they are useful for software development. The first implementation efforts have begun, using TNO's product modelling tool PMshell. In the near future, implementation and testing will play an increasingly important role in this research.

## ACKNOWLEDGEMENT

Several colleagues have contributed to the ideas presented here, most notably Wim Bakkeren, Bart Luiten and Frits Tolman. Particularly the ideas on view models and system models have been described earlier in (Tolman, 1993). The author also had benefit from discussions in the Dutch research project NOBI and the ESPRIT project ATLAS.
This research is sponsored by the Technology Foundation (STW).

## REFERENCES

Björk B.C., Pentillä H., Finne C., Nervola M., Saarinen H., Moisio J. (1989), "A Prototype Building Product Model Using a Relational Database", *Proceedings ARECDAO*, Barcelona.

Björk B.C. (1992), "A conceptual model of spaces, space boundaries and enclosing structures", *Automation in Construction,* Volume 1, Number 3, December.

Burns, T., Fong, E., Jefferson, D., Knox, R., Mark, L., Reedy, C., Reich, L., Roussopoulos, N., and Truszkowski, W. (1985), "Reference Model for DBMS Standardization," Tech. Rep., NBSIR 85-3173, May.

Eastman, C.M. (1991), "Use of Data Modeling in the Conceptual Structuring of Design Problems", *CAAD Futures '91,* Vieweg & Sohn Verlagsgesellschaft, Zürich / Braunschweig.

Gielingh, W.F. (1988), "General AEC Reference Model," Tech. Rep. BI-88-150, TNO Bouw, Rijswijk, the Netherlands, October .

Nijssen, G.M. and Halpin, T.A., (1989), *Conceptual Schema and Relational Database Design: A fact oriented approach,* Prentice Hall.

STEP (1992), *Product Data Representation and Exchange – Part 1: Overview and Fundamental Principles,* STEP document ISO TC184/SC4/PMAG.

Tolman, F., Van Nederveen S., Bakkeren W. (1993), "A Systems Approach to Product Modelling", Tech. Rep. 93-BI-R0054, TNO Bouw, Rijswijk, the Netherlands.

Turner, J. (1988), "AEC Building Systems Model", Michigan, January.

Van Dam, P.R., Eeltink, M., Plokker, W., Taal, A.C., Van Veghel, M.L.W. (1992), *DUS boek,* Tech. Rep., VABI, Delft, April (in Dutch).

# Terms and Abbreviations

**AEC** - Architecture, Engineering and Construction.

**CAD** - 1. (officially) Computer Aided Design; 2. (in practice) Computer Aided Drawing.

**CAxx -** general term for any Computer Aided application, e.g. Computer Aided Design, Computer Aided Engineering, Computer Aided Manufacturing, etc.

**Class -** a term for a group of things with common characteristics.

**Classification -** the creation of *class*es based on a specific distinctive criterion.

**Decomposition** - structuring mechanism used to describe a part-whole hierarchy, using the relationship "consists of", "is composed of", "decomposes into" or similar.

**FU, Functional Unit** - description of an *object* in terms of functional requirements, with as-required properties, as opposed to a *TS*.

**Function** - term that describes what an object is designed for, made for, or used for.

**GARM, General AEC Reference Model** - an information model for Architecture, Engineering and Construction in which some basic principles for *PDT* have been introduced, especially the distinction between *FU* and *TS.*

**HSL, High Speed Line** - 1. (general) a railway track for high speed trains. 2. (in this thesis) the new railway track for high speed trains between Amsterdam and Paris,

**ICT -** Information and Communication Technology.

**Instance -** a thing in real life, that can be counted, as opposed to a *class*.

**Instantiation** - the creation of an *instance* of a class.

**IT -** Information Technology.

**Object -** 1. (generally) thing; 2. (software engineering) basic element in object oriented software engineering, that encapsulates its properties, relationships and behaviour; 3 (in the Object Tree approach) physical thing that performs a *function*, or that belongs to the context of objects that perform a function.

**OO, Object Oriented** - approach in software engineering based on the use of *object*s as basic elements in software development.

**OT, Object Tree** - simple hierarchical list of *object*s of a project, in which object names, identification, basic properties and *decomposition* structure are stored.

**PDM, Product Data Management** - a technology that aims at the organization and maintenance of product data, with the emphasis on the representations (documents, drawings, etc.) rather than on the contents (product parts etc.).

**PDT, Product Data Technology** - a technology that aims at representation and exchange of product data in a meaningful way. See also *STEP*.

**Performance** - degree to which a solution meets the requirements.

**PM, Product Model -** an information model of a product *instance*, including information on product parts, properties, relationships, during different life cycle stages.

**Product Type** - term for a group, or *class* of similar products. E.g. "building", "road".

**PtM, Product Type Model** - an information model of a *product type*, including information on product parts, properties, relationships, during different life cycle stages.

**STEP** (officially ISO 10303 Product Data Representation and Exchange) - the emerging ISO standard for *PDT*.

**TS, Technical Solution** - description of an *object* in terms of a designed thing with as-expected properties.

**VM, View Model** - a discipline-specific information model of a product *instance.*

**VtM, View Type Model** - a discipline-specific information model of a *product type.*

# References

[Aalami and Fischer, 1998] Florian Aalami and Martin Fischer, "Joint product and process model elaboration based on construction method models", in *The Life-Cycle of Construction IT Innovations*, CIB W78 Conference Proceedings, KTH Stockholm, ISBN 91-7171-281-4, 1998.

*[ADSE 1996] ADSE B.V., Handouts Workshop Systems Engineering at HSL-Zuid.*

[Alberts & Dikker 1994] L.K. Alberts and F. Dikker, Introduction to *Workshop notes of the workshop "A Semantic Basis for Sharing Knowledge and Data in Design,* L.K. Alberts (ed), held prior to the Third International Conference on Artificial Intelligence in Design, 15-18 August 1994, Lausanne, Switzerland.

[Alexander 1964] C. Alexander, *Notes on the Synthesis of Form,* Harvard University Press, ISBN 0-674-62751-2, 1964.

[Amor & Hosking, 1993] Amor, R.W. and J.G. Hosking, "Multi-Disciplinary Views for Integrated and Concurrent Design," in proceedings *Management of Information Technology for Construction*, in Singapore, eds. K.S. Mathur, M.P. Betts, and K Wai Tham, pp. 255-267, World Scientific Publishing Co. Pte. Ltd., 1993 (CIB W78).

[Anderl & Arlt, 1999] Reiner Anderl, Martin Arlt, DiK Darmstadt University of Technology, "iPDM Systems", in: *Proceedings: PDT Days 1999,* Stavanger, Norway, 1999..

[Angus 1994] Chris Angus, *EPISTLE Framework*, Angus Associates, UK, 1994.

[ATLAS 1993] ATLAS consortium, *ATLAS Public Project Overview*, 1993.

[Augenbroe] G.L.M. Augenbroe (ed) *COMBINE 1 Final report,* CEC-Joule report, Delft University, The Netherlands, 1993.

[Augenbroe] G.L.M. Augenbroe (ed) *COMBINE 2 Final report,* CEC-Joule report, Delft University, The Netherlands (http://dutcu15.tudelft.nl/~combine/), 1995.

[Augenbroe cs 1998] G. Augenbroe, W. Rombouts and M. Verhoef, "Product Data Technology in Integrated A/E/C/ Systems: Past, Present and Future", in R. Amor (ed), *Product and Process Modelling in the Building Industry,* British Research establishment, Watford, UK, 1998.

[Bakkeren 1997] W. Bakkeren, *Computer Integrated Structural Engineering,* PhD-thesis, Delft University of Technology, 1997.

[Björk 1995] B-C Björk, "Requirements and Information Structures for Building Product Data models", doctoral thesis, KTH Stockholm, 1995.

[Brussaard 1980] B.K. Brussaard, T.A. Tas, "Information and Organization Policies in Public Administration", in S.H. Lavington (ed.), *Information Processing 80,* North-Holland, 1980.

[Dietz 1996] J.L.G. Dietz, *Introductie tot DEMO,* Samson, Alphen aan den Rijn, 1996.

[EDIFACT 1988] ISO, *Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) – Application Level Syntax Rules"* ISO 9735, Geneva 1988.

[Gielingh 1988] W. Gielingh, *General AEC Reference Model,* TNO Report BI-88-150.

[Hannus cs 1995] M. Hannus, K. Karstila and V. Tarandi, "Requirements on standardized building product data models". In *Product and process modelling in the building industry,* R. Scherer (ed), Dresden, 1994.

[IAI], International Alliance for Interoperability, "Industry Foundation Classes", Release 1.5, 1997.

[INCOSE 1998] International Council on Systems Engineering (INCOSE), *"Systems Engineering Handbook"*, January 1998.

[IOP Bouw 1989] IOP Bouw "Bouw Informatie Model", Stichting Bouwresearch, Rotterdam, 1989.

[ISO 1993] ISO/TC184, "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), 1993.

[ISO 1994] ISO/TC184, “Part 106: Building Construction Core Model,” eds. J. Wix, F.P. Tolman, P. Poyet, and B.C. Björk, Project Proposal, ISO TC184/SC4/WG3 N106, 1994 (SC 4).

[Junge and Liebich 1998] Junge, R. and Liebich, T. “Product Modelling Technology – the Foundation of Industry Foundation Classes”, in R. Amor (ed), *Product and Process Modelling in the Building Industry,* British Research establishment, Watford, UK, 1998.

[Latham 1994] M. Latham, *Constructing the Team,* HMSO, United Kingdom.

[Luiten 1994] G.T. Luiten, *Computer Aided Design for Construction in the Building Industry,* PhD Thesis, Delft University of Technology, 1994.

[MARITIME 1995] Modelling And Reuse of Information over TIME. ESPRIT 6041, July 1992-October 1995.

[Nederveen, van 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, 1993 (CIB W78).

[Nederveen, van 1994] Nederveen, S. 1994. ATLAS Deliverable D106-III, View type Model for Architecture.

[Nederveen, van 1995] Sander van Nederveen (editor), IDM, HTML-version, available through WWW at: `dutcu15.tudelft.nl/~combine/idm`.

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

[PLib 1995] ISO TC 184/SC4 *ISO 13584 Parts Library,* Geneva, Switzerland, 1995.

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

[Ridder, de 1999] H.A.J. de Ridder, Contribution to PAO Workshop *Design-Construct,* Delft University, Faculty of Civil engineering.

[RISMAN 1998] J. Blazer and D. Stam (ed), *De RISMAN-methode - Een instrument voor het risicomanagement van grote infrastructuurprojecten*, participants: Gemeentewerken Rotterdam, NS Railinfrabeheer, Rijkswatsrstaat, TU Delft, Twijnstra Gudde, available at CUR, Gouda, the Netherlands, ISBN 90744111 IOX, 1998.

[Rosenman et al 1993] Rosenman, M.A., J.S. Gero, and Y-S Huang, "Representation of Multiple Concepts of a Design Object Based on Multiple Functions," in proceedings *Management of Information Technology for Construction*, in Singapore, eds. K.S. Mathur, M.P. Betts, and K Wai Tham, pp. 239-254, World Scientific Publishing Co. Pte. Ltd., 1993 (CIB W78).

[Rush 1986] Rush, R.D. (ed.) *Building Systems Integration Handbook*, Wiley, New York, USA, 1986.

[Scherer 1994] R. Scherer "EU-project COMBI - objectives and overview", in: . In *Product and process modelling in the building industry,* R. Scherer (ed), Dresden, 1994.

[Sol 1988] H. Sol, *Information Systems Development: a problem solving approach*, Delft University of Technology, 1988.

[Spekkink 1992] D. Spekkink, *Programma van eisen, instrument voor kwaliteits-beheersing,* SBR-publicatie 258, ISBN 90-5367-056-4, Stichting Bouwresearch, Rotterdam 1992.

[Staub & Grabowski 1999] G. Staub and H. Grabowski, "Components-based Product Data Models: the Future of Data Modelling", in: *Proceedings: PDT Days 1999,* Stavanger, Norway, 1999.

[Tarandi 1998] Tarandi, V. Neutral Intelligent CAD Communication (information exchange in construction based upon a minimal schema), PhD thesis, KTH, Sweden, 1998

[Teigeler 1996] H. Teigeler, *STEP AP 221 Handbook*, November 1996.

[Tolman 1994a], F.P. Tolman, ATLAS Deliverable D106-I, LSE Project type Model, 1994.

[Tolman 1994b] Tolman, F. P. ATLAS Deliverable D106-II, Building Project type Model, 1994.

[Tolman 1995] Frits Tolman, Sander van Nederveen, *Building Project type Model ,* Version 1.0, ESPRIT project 7280 (ATLAS), Deliverable D106-III.

[Tolman 1998] paper presented at the CONCUR workshop in Amsterdam in 1998

[Tolman 1999] Tolman F. P. Product modeling standards for the building and construction industry: past, present and future. Automation in Construction. Februari 1999.

[Turner 1988] Turner, J.A., “A systems approach to the conceptual modeling of buildings,” in proceedings *Conceptual Modelling of Buildings*, in Lund, Sweden, eds. Per Christiansson and Henry Karlsson, pp. 179-187, Swedish Building Centre, 1988 (CIB W78 and W74).

[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, 1998.

[Veld, In ‘t, 1983] Prof. Ir. J. In ’t Veld, *Analyse van organisatieproblemen,* Elsevier Amsterdam/Brussel, 1983.

[VISI 1999] Projectgroep VISI, *VISI Eindrapport Onderzoeksfase, Communicatie in projecten uitdagingen voor de GWW-sector,* LWI, Gouda, 1999.

[Vries, de 1991] B. de Vries, “The Minimal Approach”, in: *Proceedings CIB W78, Eindhoven University, 1991.*

[Vries, de 1996] B. de Vries, “*Communication in the Building Industry*”, PhD-thesis, Eindhoven University, ISBN 90-6814-538-X, 1996.

[Waard, de 1992] M. de Waard, *Computer Aided Conformance Checking,* PhD-thesis, Delft University, 1992.

[West, 1994] Matthew West, *Developing High Level Data Models,* Shell UK, 1994.

[Willems, 1998] Peter Willems, *Conceptual Modelling of Structure and Shape of Complex Civil Engineering Projects,* PhD-thesis, Delft University, ISBN 90-9011888-8, 1998.

[Wix and Liebich 1998] J. Wix and T. Liebich, “Industry Foundation Classes Some Business Questions Examined”, in R. Amor (ed), *Product and Process Modelling in the Building Industry,* British Research establishment, Watford, UK, 1998.

[Womack et al, 1991] Womack, James, Jones, Daniel, & Roos, Daniel, *The machine that changed the world.* New York: Harper-Collins, ISBN 0-06-097417-6.

[Woestenenk 1999] K. Woestenenk, “The LexiCon”, CEC ’99, 1999. To appear.

[Zarli cs 1999] A. Zarli, E. Buckley, O. Richaud, "A Componentware approach for the new generation of Business Applications in Construction", CEC '99, 1999. To appear.

# Curriculum Vitae

| | |
|---|---|
| 22 June 1961 | Born in Hengelo (O), The Netherlands. |
| 1973-1979 | Atheneum B at Rijnlands Lyceum Oegstgeest. |
| 1980-1988 | Study Architecture at Delft University of Technology.<br>• *MSc-thesis: "Voorstudie voor een Geïntegreerd Gebouw-model" (Preliminary Study on an Integrated Building Model).* |
| 1988-1990 | Employed at Modern Medium (a CAD-consultancy firm) |
| 1990-present | Employed at TNO, Department of Information Technology for Building and Construction.<br>• *Task leader "View type Model for Early Architectural Design" for the ESPRIT-project ATLAS (1992-1995).*<br>• *Task leader "Integrated Data Model" (IDM) for the EU-JOULE project COMBINE (1992-1995).* |
| 1991-present | PhD-research at Delft University. |
| 1997-1999 | Hired by the High Speed Line project (HSL-Zuid).<br>• *Responsible for Configuration Management: product data management, design change control, interface management and risk management.* |

# Summary

## 1. Communication in Building and Construction

The building and construction industry is facing great challenges. A large amount of work is waiting, 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 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.

## 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. For example by taking care that risks are managed by the party which is best equipped for the job.

## 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), 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, see below.

## 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.

## 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. 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 also plays a 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 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).

But CAD systems and CAD data are fundamentally limited because the underlying work process, the process of drawing, has not changed. Building product data is represented as geometric data. The meaning of the geometric data is defined implicitly, and must be retrieved by human interpretation. A collection of lines is only recognized as a column by interpretation of an expert in building.

Already a long time ago researches started to look for a better approach in which design information could be represented and exchanged in a meaningful way. In this way a column is not just represented as a collection of lines, but as an object with a name "column" and with associated data on shape, material etc. This is the basic idea of what nowadays is called product modelling. Subsequently, a need emerged for standardization of information structures that could support such meaningful representation and exchange. This resulted in the mid eighties in the initiative to the ISO-STEP standard, officially ISO 10303.

The goal of STEP was (and is) a standard for representation and exchange of product data in a semantic manner: not just geometry but also data on material, connections, product structure, structural and thermal properties, etc. The STEP development was not specificly directed to building and construction, but it resulted in a number of follow-up initiatives and projects for various industries, including building and construction. Of course the STEP development has been closely related to developments in ICT, for example about database concepts.

Until now, the goal of STEP is only partially achieved. There is a generic basis (the so-called Generic Resources), but these are still subject of discussion. For other industries than building and construction, specific STEP-based standards have been developed, for example for the process industry, for shipbuilding and for the automotive industry. But these standards are developed in different ways, and they are not compatible with each other, despite their common STEP-basis.

For building and construction a widely accepted standard based on STEP is still missing. But why? Why are we unable to develop an internationally accepted, powerful standard for electronic communication in building and construction? Several factors play a role:

- 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). These IFCs are essentially standardized CAD objects that contain both geometric and semantic product data. The IAI that is developing the IFCs, is a consortium of CAD vendors, such as AutoDesk, Bentley, Nemetschek, etc. 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

In order to come to an approach for better communication and information exchange in building and construction, several developments can be taken into account: (1) trends in basic technologies and (2) new concepts for communication in building and construction.

### *Basic Technologies*

With respect to trends in basic technologies, we will first take a look at developments in STEP. Within STEP several modelling languages have been developed, most notably EXPRESS, and a number of implementation principles have been elaborated. The basic principles of STEP are sound: data exchange on a semantic level using an open standard. But in building and construction, the STEP approach has not been successful. The standardization process is moving very slowly, partly because of a continuous lack of financial means. Furthermore, the used ICT concepts turned out to be ineffective and outdated, especially compared to newer approaches for ICT.

This has led to a number of additions and alternatives. For example the support of behaviour using the object-oriented CORBA technology. 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.

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

But the most promising trend in the context of basic technologies, is the ongoing development of Internet technology, especially with XML.

### *New Concepts for Communication*

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. 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 also elaborated in different ways. For building and construction the approach is worked out most extensively in the European project ATLAS (1991-1994). 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.

## 4. Research Question

Looking back at the various efforts, there is no doubt that the STEP approach for neutral semantic representation and exchange of product model data is a sound approach. Moreover, it is a prerequisite for better (electronic) communication in building and construction. However, in order to achieve practical results, some huge pitfalls exist:

1. getting lost in too complex information models, leading to disappointing results in implementation and usage,
2. unfortunate choice of information technology that becomes outdated because of newer developments,
3. too much expectations from standardization developments.

Then how can we achieve better communication avoiding these pitfalls? As said above, neutral semantic representation and exchange of product model data is regarded as a prerequisite. Solutions based on CAD systems will not really bring the building and construction industry forward. Furthermore, the emphasis must be on simplicity. Finally it is useful to emphasize on project management support, since project managers can stimulate the implementation of new communication concepts.

This leads to the following research question:

*"How to develop an approach for interdisciplinary communication in large-scale building and construction projects that is neutral, pragmatic and bottom up, primarily serving the needs of (one of) the most influential participants in a project?"*

# 5. Object Trees

In order to answer the research question above, we have to act pragmatically and bottom-up. This implies that a number of "STEP habits" must be reconsidered. This has led to an approach which we will call the *Object Tree approach.*

The main characteristics of the Object Tree approach are listed below:

- An Object Tree is an instance model,
- Objects are function performers,
- An Object Tree is a decomposition tree,
- An Object Tree contains a minimum set of relationships,
- For shape description a reference to CAD drawings is specified.

## An Object Tree is an Instance Model

Or in common English: an Object Tree describes a single thing, not a class of things.

One of the "STEP habits" stated above is to start with a product type model, a model that describes a certain class of objects, such as buildings, roads or viaducts. Instead, the Object Tree approach starts with the occurrence. In other words: an Object Tree is an instance model, not a type model.

In this way we can prevent that (1) the development of type models becomes a very slow and tedious progress, and (2) that the development of instance models gets under pressure, leading to disappointing results. This was one of the major problems in STEP-related research in the early nineties.

## 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 an Object Tree can be regarded as a solution tree, in which also the functions are specified for which the objects are solutions.

## An 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. But the 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.

### *An Object Tree Contains a Minimum Set of Relationships*

Relationships between objects is a discussion item that can easily lead to a very complex model structure. For that reason an 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. And therefore no functional, logical or any other kind of relationships.

### *For Shape Description a Reference to CAD Drawings 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.

## 6. Implementation of Object Trees

As the Object Tree is basically rather simple, the implementation of 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.

## 7. The HSL Case

In the Dutch High Speed Line (HSL) project the approach described above has been applied in the so-called HSL Object Tree. This 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, sound barriers, cables and ducts etc.

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

From the HSL project it can be concluded that even a simple approach can easily become too difficult. One might think that the decomposition structure has been defined in a couple of days. In reality this has taken several months, including many meeting hours from many people. Also on the implementation side a simple step sometimes took months to take.

Nevertheless the HSL Object Tree can be regarded as a useful contribution to the enhancement of communication in a large scale building and construction project based on semantic product data.

# 8. Conclusions

The main conclusions 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 too complex information models, unfortunate ICT choices and disappointing developments in standardization.
3. By using 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 an Object Tree must meet the following requirements:

- The Object Tree must be developed bottom-up, i.e. objects (instances) first, classes later,
- 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.
- The Object Tree must be neutral, i.e. independent of software vendors, building participants and standardization efforts.
- Shape description must be taken care of by a reference to CAD drawings.

  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,
- Development of management methods using the Object Tree, for example interface management and risk management,
- Generalization of the Object Tree towards a type model.

# Samenvatting

## 1 Communicatie in de bouw

De bouw staat de komende jaren voor grote uitdagingen. Er ligt een grote hoeveelheid werk te wachten, inclusief een aantal zeer grote projecten. In Nederland bijvoorbeeld de uitbreiding van Schiphol, de tweede Maasvlakte, de Betuweroute, de HSL en andere grote spoorprojecten. Bij dergelijke projecten, maar ook bij kleinere, worden steeds hogere eisen gesteld aan de beheersing van kwaliteit, tijd en kosten.

Van oudsher wordt de bouw gekenmerkt door wisselende samenwerkingsverbanden tussen verschillende disciplines van verschillende organisaties. Daarbij is de communicatie tussen disciplines een kritische succesfactor. Bij het streven naar betere beheersing van kwaliteit, tijd en kosten richt men zich dan ook in belangrijke mate op verbetering van de onderlinge communicatie.

De laatste jaren zijn een aantal ontwikkelingen te herkennen die zijn gericht op het verbeteren van de communicatie tussen disciplines, zoals:

- nieuwe contractvormen, Design & Construct, Turnkey, e.d.,
- classificatie en codering,
- prestatie-benadering,
- systeembenadering.

## *Nieuwe contractvormen*

Op dit moment staan nieuwe contractvormen voor aanbesteding zeer in de belangstelling, zoals Design & Construct en Built - Operate - Transfer. De achterliggende gedachte is om de capaciteiten van de verschillende partijen zo goed mogelijk te gebruiken. O.a. door ervoor te zorgen dat oplossingen worden ontwikkeld en risico's worden beheerst door de partijen die dit het beste kunnen.

## *Classificatie en codering*

Classificatie, of *identificatie* is het onderscheiden van (object)klassen; codering is het toekennen van codes aan deze klassen. Het bekendste voorbeeld is SfB. Oorspronkelijk richtte classificatie in de bouw zich vooral op gebouw-elementen. Later is men classificaties gaan uitbouwen met bijv. werksoorten naast elementen (zie STABU), met verschillende decompositieniveaus, met bibliotheekstructuren, etc. In wezen gaat het daarmee niet meer over classificatie maar over gebouwmodellering, zie verder.

## *Prestatie-benadering*

Bij de prestatiebenadering gaat het erom dat de bouwopgave wordt omschreven in termen van kwantificeerbare, geëiste en gewenste prestaties, i.p.v. in termen van voorgeschreven oplossingen.

## *Systeembenadering*

De systeembenadering houdt eenvoudig gezegd in dat het te leveren product wordt beschouwd als een verzameling systemen, die een bepaalde prestatie moeten leveren. Bijvoorbeeld een ruimte-systeem, een draagsysteem of een verwarmingsssysteem.

Ontwikkelingen als hierboven omschreven hangen sterk met elkaar samen. Bijvoorbeeld: Design & Construct vraagt om een prestatie-gericht programma van eisen; en het ligt voor de hand om hierbij uit te gaan van eisen aan te onderscheiden systemen.

# 2 ICT in de bouw

Bij communicatie in de bouw speelt uiteraard ook informatie- en communicatietechnologie (ICT) een belangrijke rol. Als de stand van zaken m.b.t. ICT in de bouw wordt beschouwd, dan kan worden vastgesteld dat deze voornamelijk is gebaseerd op CAD-systemen (tekensystemen) en uitwisseling van CAD-data. Daarbij is, in zeer bescheiden mate, sprake van integratie met CAE-systemen (rekenprogramma's), besteksprogramma's etc.

CAD-systemen en CAD-data zijn echter fundamenteel beperkt doordat het onderliggende werkproces, het tekenen, niet anders is dan vroeger. Bouwkundige informatie wordt weergegeven als geometrische informatie. De betekenis hiervan is impliciet vastgelegd en moet worden teruggevonden door menselijke interpretatie. Een verzameling lijntjes op een tekening wordt slechts door interpretatie van een bouwkundige herkend als een kolom.

Al geruime tijd geleden is men gaan zoeken naar een betere benadering waarbij ontwerpgegevens betekenisvol kunnen worden vastgelegd en uitgewisseld. Daarbij wordt een kolom niet als een groep lijntjes beschreven, maar als een object met de naam "kolom" en verder met vorm, materiaal, etc. Dit idee is de achtergrond van wat nu productmodelleren wordt genoemd. Het besef dat de gebruikte begrippen uniek gedefineerd zouden moeten zijn leidde vervolgens in de jaren tachtig tot de ontwikkeling van de STEP-standaard, officieel ISO 10303.

Het doel van STEP was (en is) het komen tot een standaard voor het vastleggen en uitwisselen van betekenisvolle productgegevens: niet alleen geometrie maar ook gegevens over materiaal, verbindingen, productstructuur, constructieve en fysische eigenschappen etc. Dit initiatief richtte zich niet specifiek op de bouw, maar mondde uit in een reeks van initiatieven en projecten voor verschillende industrieën. Uiteraard hangt de STEP-ontwikkeling sterk samen met ICT-concepten en ontwikkelingen hierin.

Tot dusverre is het doel van STEP maar gedeeltelijk bereikt. Er is een generieke basis (de zgn. generic resources) waar echter de nodige discussiepunten over bestaan. Voor andere industrieën dan de bouw zijn specifieke standaards ontwikkeld, bijvoorbeeld voor de procesindustrie, de scheepsbouw en de auto-industrie. Deze standaards zijn verschillend in opzet en mede daardoor niet compatibel met elkaar, ondanks het bestaan van de generieke basis.

Voor de bouw is er na al die jaren nog steeds geen algemeen aanvaarde en gebruikte STEP-standaard. Hoe komt dat? Waarom lukt het niet om een internationale, breed gedragen, krachtige standaard voor de bouw te ontwikkelen? Verschillende factoren spelen een rol:

- het veelal kleinschalige, lokale dan wel nationale karakter van de bouwbranche, zonder dominante partijen, waardoor het moeilijk is om te komen tot branche-afspraken,
- het internationale karakter van de IT-branche waardoor het extra moeilijk is om te komen tot IT-ondersteuning van standaards voor de bouw,
- het bestaan van diverse IT-benaderingen voor ondersteuning van de standaard, en (vooral) het steeds weer ontstaan van nieuwe benaderingen.

Een recentere ontwikkeling op dit gebied is de ontwikkeling van Industry Foundation Classes (IFCs) door de International Alliance for Interoperability (IAI). Dit zijn in wezen gestandaardiseerde CAD-objecten die zowel geometrische als betekenisvolle productinformatie bevatten. De IAI die deze ontwikkelt is een consortium van CAD-leveranciers zoals Autodesk, Bentley, Nemetschek e.a. Deze ontwikkeling heeft duidelijk voordelen boven de STEP-ontwikkelingen, o.a. doordat deze getrokken wordt door de CAD-leveranciers. Maar ook de IFCs komen uiterst langzaam tot stand.

## 3 Uitgangspunten voor verbetering

Bij het zoeken naar een benadering om communicatie en informatieuitwisseling te verbeteren kan worden gekeken naar verschillende ontwikkelingen: (1) basistechnologieën, en (2) nieuwe concepten voor communicatie in de bouw,

### *Basistechnologieen*

Voor de basistechnologieën komen we in eerste instantie uit bij STEP. Binnen STEP zijn modelleertalen ontwikkeld (in het bijzonder EXPRESS), en zijn verschillende implementatieprincipes uitgewerkt. Het uitgangspunt van STEP is goed: betekenisvolle gegevensuitwisseling met behulp van een open standaard. In de uitwerking voor de bouw schiet STEP echter tekort. Het standaardisatieproces verloopt uiterst moeizaam, mede door het voortdurende gebrek aan capaciteit en financiële middelen. Daarnaast blijkt de IT-basis niet (meer) de meest effectieve, en worden ontwikkelingen achterhaald door nieuwe IT-benaderingen.

Dit heeft geleid tot verschillende alternatieven en aanvullingen. Bijvoorbeeld het ondersteunen van functies en gedrag met de objectgeorienteerde CORBAtechnologie. In het ESPRIT-project VEGA is recentelijk gewerkt aan de toepassing van CORBA in combinatie met STEP voor ondersteuning van werkstroombeheersing (workflow management) in bijvoorbeeld de bouw.

Een andere trend is aan te duiden als "the minimal approach". In deze benadering wordt het gegevensmodel tot een minimum beperkt om zo te komen tot een handzaam formaat voor gegevensuitwisseling in de bouw. Uitwerkingen van deze benaderingen hebben zich gericht op gebouwgeometrie (Tarandi) en op integratie met EDI (de Vries).

De meest belovende ontwikkeling in de sfeer van basistechnologieën is echter de verdere ontwikkeling van Internettechnieken, in het bijzonder met XML.

Wat betreft nieuwe concepten voor communicatie in de bouw is de zgn. view-benadering van belang. Deze benadering gaat uit van de vaststelling dat partijen in de bouw verschillende disciplines vertegenwoordigen die elk een eigen invalshoek op ontwerpinformatie hebben. Als gevolg daarvan hebben zij elk een eigen informatiebehoefte, die ondersteund moet worden met specifieke gegevensverzamelingen (te beschrijven in zgn. view-modellen). Voor communicatie in de bouw betekent dit dat view-conversie ondersteund moet worden.

Ook de viewbenadering is op verschillende manieren uitgewerkt. Voor de bouw het verst in het ESPRIT-project ATLAS (1991-1994). Het grote probleem bij de view benadering, zoals in het ATLAS-project geconstateerd, is dat de relaties tussen de diverse, view-afhankelijke informatiemodellen (de view-conversie) te ingewikkeld worden, waardoor implementatie en onderhoud van de conversie-software te kostbaar wordt.

## 4 Probleemstelling

Terugkijkend kan worden vastgesteld dat de STEP-benadering voor neutrale, betekenisvolle gegevensuitwisseling op zich een goede benadering is, en noodzakelijk om communicatie in de bouw verder te brengen. Maar bij het komen tot toepasbare resultaten bestaan enkele grote valkuilen:

(1) verzanden in te ingewikkelde informatie-modellen, waardoor van implementatie en toepassing weinig terecht komt.

(2) kiezen voor een informatie-technologie die na verloop van tijd achterhaald wordt door nieuwe ontwikkelingen,

(3) te grote verwachtingen van standaardisatie, branche-afspraken e.d.

### *Hoe dan wel?*

Hoe dan wel? Zoals gezegd is het uitgaan van neutrale, betekenisvolle productinformatie een voorwaarde voor verbetering. Met oplossingen gebaseerd op CAD-systemen alleen komen we er niet. Maar verder moet vooral worden gestreefd naar eenvoud. Tenslotte is het zinvol om te streven naar ondersteuning van het projectmanagement; het projectmanagement kan er immers voor zorgen dat nieuwe communicatieconcepten snel ingevoerd kunnen worden.

De vraag die dit onderzoek wil beantwoorden is daarom:

*Hoe kan een benadering voor interdisciplinaire communicatie in grote bouwprojecten worden ontwikkeld, die neutraal, pragmatisch en bottom up is en in elk geval zorgt voor ondersteuning van het projectmanagement?*

# 5 Objectenbomen

Om de hierboven gestelde onderzoeksvraag te beantwoorden, is het zaak pragmatisch en bottom up te werk te gaan. Hierbij zijn een aantal "automatismes" van STEP-gerelateerde projecten ter discussie gesteld. Dit heeft geleid tot een benadering die we zullen aanduiden als de objectenboom-benadering.

Hieronder worden puntsgewijs de belangrijkste kenmerken van de objectenboom-benadering aangegeven.

- *Een objectenboom is een instantie-model*
- *Objecten zijn functie-vervullers*
- *De objectenboom is een decompositieboom*
- *De objectenboom beperkt zich tot een minimum aan relaties*
- *Voor vormbeschrijving wordt verwezen naar CAD-tekeningen*

### *Een objectenboom is een instantie-model*

Of in gewoon Nederlands: een objectenboom beschrijft een enkel ding, geen klasse van dingen.

Eén van de hierboven aangeduide "STEP-automatismes" is het beginnen bij een zgn. type-model, een model dat een bepaalde klasse van objecten beschrijft, bijvoorbeeld gebouwen, wegen of viaducten. De objectenboom-benadering gaat echter uit van het exemplaar. Met andere woorden: de objectenboom is een instantiemodel, geen typemodel. Hierdoor wordt voorkomen dat de ontwikkeling van het typemodel zo langdurig en ingewikkeld wordt dat de ontwikkeling van instantie-modellen niet uit de verf komt. Dit was één van de grote problemen bij de STEP-gerelateerde projecten in het begin van de jaren '90.

## *Objecten zijn functie-vervullers*

De "bouwstenen" van een objectenboom zijn de objecten. Als objecten worden beschouwd: fysieke dingen die een functie vervullen. Deze functie kan zijn het leveren van een bepaalde geëiste prestatie, vastgelegd in een programma van eisen.

Op deze wijze is de objectenboom te beschouwen als een oplossingsboom, waarbij tevens wordt vastgelegd voor welke functies de objecten een oplossing zijn.

## *De objectenboom is een decompositieboom*

Zoals de aanduiding "boom" suggereert heeft de objectenboom een hiërarchische structuur. Nu zijn er vele manieren om hiërarchie aan te brengen in een productstructuur. De objectenboombenadering streeft echter naar eenvoud en kiest daarom voor één hiërarchisch principe: decompositie ("bestaat-uit").

Echter, ook met decompositie kan met nog verschillende kanten uit. De twee basisprincipes zijn:

- subsysteem- decompositie, dwz decompositie naar vorm en plaats,
- aspectsysteem-decompositie, dwz decompositie in objecten met een gemeenschappelijke rol.

Beide decompositieprincipes zijn nodig en maken daarom deel uit van de objectenboombenadering. De relatie tussen beiden is echter complex, en wordt daarom niet vastgelegd in de objectenboom.

*De objectenboom beperkt zich tot een minimum aan relaties*

Relaties tussen objecten is wederom een onderwerp dat kan leiden tot complexe modelstructuren. Daarom wordt voor objectenbomen een minimum aan soorten relaties gehanteerd. Allereerst de twee soorten decompositie-relaties als hierboven omschreven. Daarnaast alleen nog fysieke raakvlakken. En dus geen functionele, logische of andersoortige relaties.

*Voor vormbeschrijving wordt verwezen naar CAD-tekeningen*

Ook voor het opnemen van vormbeschrijving bestaan vele methoden, die echter meestal ook leiden tot zeer complexe modellen. Ook hierbij wordt daarom gekozen voor eenvoud, en wel door te verwijzen naar CAD-tekeningen.

## 6 Implementeren van objectenbomen

Door de eenvoud van de benadering kan ook de implementatie van de objectenboom met ondersteunende software eenvoudig worden gehouden. Het kan zelfs met systemen als Excel en Access, maar levert dan weinig ondersteuning op het gebied van gegevensbeheer en -onderhoud, c.q. user interface. Een beter maar duurder alternatief is één van de vele PDM-systemen die op de markt zijn. In de nabije toekomst zullen vooral nieuwe besturingssystemen en Internet-software de mogelijkheden voor softwareondersteuning verder vergroten.

## 7 De HSL Case

Bij het HSL-project is de hierboven beschreven benadering in grote lijnen toegepast in de zg HSL-objectenboom. Dit heeft geleid tot een boomstructuur met bovenin uiteraard de HSL-spoorlijn. Deze decomponeert in een paar stappen tot enkele duizenden HSL-objecten, zoals bruggen, viaducten, duikers, geluidsschermen, te verleggen kabels en leidingen etc.

Eigenlijk is de HSL-benadering nog eenvoudiger geweest dan de hier gepropageerde benadering. Zo zijn o.a. de functionele kant van objecten, de aspectsysteem-decompositie en het vastleggen van fysieke raakvlakken in de HSL maar gedeeltelijk uitgewerkt. De implementatie van de HSL-objectenboom is gedaan met achtereenvolgens Excel, Access en het PDM-systeem SmarTeam.

Uit het HSL-project kan worden geconcludeerd dat ook een simpele aanpak al snel te moeilijk is. Wellicht is de indruk ontstaan dat de decompositiestructuur in een dag of wat tot stand is gekomen. In werkelijkheid zijn er vele uren van vele mensen in gaan zitten - vooral vergaderuren. Ook t.a.v. implementatie bleek een voor de hand liggende conclusie soms maanden te vergen.

Toch kan de HSL-objectenboom al met al worden beschouwd als een bijdrage aan betere communicatie in een groot bouwproject op basis van betekenisvolle product/objectgegevens.

# 8 Conclusies

De belangrijkste conclusies zijn:

1. Voor verbetering van de communicatie in de bouw is het betekenisvol vastleggen en uitwisselen van productgegevens een voorwaarde.
2. De ontwikkeling van methoden en technieken voor betere communicatie in de bouw mislukt gemakkelijk door te ingewikkelde informatiemodellen, ongelukkige ICT-keuzes en tegenvallende standaardisatie-ontwikkelingen.
3. Door uit te gaan van een zgn. objectenboom kan in relatief korte tijd een eenvoudige maar complete gegevensstructuur voor een bouwproject worden ontwikkeld, en daarmee een basis voor betere communicatie.
4. Zo'n objectenboom moet voldoen aan de volgende kriteria:
   - bottom-up-ontwikkeling: eerst de objecten (instanties), later de klasses,
   - objecten als fysieke dingen die een functie vervullen,
   - een decompositiestructuur, met zowel subsysteem-decompositie (naar plaats) als aspectsysteem-decompositie (naar rol); geen andere hiërarchische structuren,
   - verder slechts één soort relatie: de fysieke raakvlak-relatie,
   - vormbeschrijving door verwijzing naar CAD-tekeningen.

   Zo'n objectenboom is relatief snel te ontwikkelen, en bovendien eenvoudig en snel te implementeren, desgewenst in commercieel verkrijgbare PDM-software.

5. De objectenboom-benadering kan als volgt worden uitgebouwd:

- (verder) ontwikkelen van classificatieafspraken voor objectnamen en objecttypes,
- (verder) ontwikkelen van bibliotheken van standaard-objecten, maar ook van standaard productiemiddelen en processen.
- ontwikkelen van management-methoden op basis van de objectenboom, zoals interface-management en risico-management,
- generalisatie van de objectenboom tot een type-model.