VTT PUBLICATIONS 245

Requirements and information structures for building product data models

Bo-Christer Björk

VTT Building Technology

Dissertation for the degree of Doctor of Technology to be presented with due permission for public examination and debate in the Auditorium R4 at the Helsinki University of Technology (Espoo, Finland) on the 8th of December 1995, at 12 o'clock noon.

TECHNICAL RESEARCH CENTRE OF FINLAND
ESPOO 1995

ISBN 951-38-4783-7 (soft back ed.)
ISSN 1235-0621 (soft back ed.)

ISBN 951-38-6030-2 (URL: http://www.inf.vtt.fi/pdf/)
ISSN 1455-0849 (URL: http://www.inf.vtt.fi/pdf/)

Copyright © Valtion teknillinen tutkimuskeskus (VTT) 1995

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, Espoo, PL 2000, 02044 VTT
puh. vaihde (09) 4561, telekopio 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, Esbo, PB 2000, 02044 VTT
tel. växel (09) 4561, telefax 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, Espoo, P.O.Box 42,
phone internat. + 358 0 4561, telefax + 358 0 456 4374

VTT Rakennustekniikka, Lämpömiehenkuja 2, Espoo, PL 1800, 02044 VTT,
puh. vaihde (09) 4561, telekopio 456 6251

VTT Byggnadsteknik, Lämpömiehenkuja 2, Esbo, PB 1800, 02044 VTT
tel. växel (09) 4561, telefax 456 6251

VTT Building Technology, Lämpömiehenkuja 2, Espoo, P.O. Box 1800, 02044 VTT, phone internat.
+ 358 0 4561, telefax + 358 0 456 4374

Technical editing Kerttu Tirronen

VTT OFFSETPAINO, ESPOO 1995
Electronic version 2003

Björk, Bo-Christer. Requirements and information structures for building product data models. Espoo 1995, Technical Research Centre of Finland, VTT Publications 245. 88 p. + app. 75 p.

UCD 69.001.3:681.3:69.002
69.003:006

Keywords construction, product models, computers, integration, utilization, standardization, construction management, requirements, comparison, object programs, theses

ABSTRACT

The term computer-integrated construction (CIC) is often used to describe a future type of construction process characterised by the extensive use of information technology. The key to successful CIC is the comprehensive integration of currently isolated computing applications in different phases of the construction process. Among the several types of data exchange standards needed to support such integration, the standards for structuring the information describing buildings (building product data models) are particularly important. No fully operational building product data models have as yet been formally standardised either on the national or international level, but the topic has been a subject of intensive research during the last few years. Building product data model proposals are usually defined using object-oriented information modelling techniques.

The research which is presented in this summarising thesis was carried out primarily during the years 1988-92 at the Technical Research Centre of Finland. The report begins with a brief introduction to the general background of research concerning CIC and building product data models. Fundamental concepts of object orientation and product modelling are explained in a separate chapter. In order to position the author's research results, the "state of the art" in this research field is briefly reviewed.

The research results are presented against the background of a kernel-aspect model framework, in line with current thinking among several leading researchers in this field. The results can loosely be classified into three distinctive groups: a number of requirements which building product data models should fulfil; specific information structures in building product data models; and the integration of product models with other types of information used in the construction process. The specific information structures which were studied include the abstraction hierarchies used in building product data models, the type object mechanism and information structures needed for modelling spaces and enclosing objects.

The report ends with a discussion of the results, comparing them with the proposals and results of other researchers. Some directions for further research are also outlined.

PREFACE

This thesis reports on the author's research concerning building product data models during the years 1988-92. It consists of five earlier published articles and this summarising report. The actual research was carried out while the author was employed as a researcher at the Laboratory of Urban Planning and Building Design of the Technical Research Centre of Finland.

The funding for the research was provided by the Technical Research Centre of Finland (VTT), the Technology Development Centre (TEKES), a number of the professional associations via the RATAS committee, and the Energy Department of the Ministry of Trade and Industry.

I wish to thank some of the numerous persons who have contributed to the work over the years. Professor Juhani Kiiras has been the supervisor of the thesis. Several colleagues at VTT were directly involved in the research projects which resulted in this thesis. In particular I would like to mention Hannu Penttilä, Kari Karstila, Sven Hult, Raine Talonpoika, Paavo Tiainen, Matti Hannus and Lauri Koskela. Of all the experts from industry who provided input and feedback Christer Finne and Jarmo Laitinen have been particularly helpful. Prof. Raimo Salokangas and Prof. Asko Sarja have both helped in creating the opportunity for this thesis. Matti Pöyry and Markku Salmi have both had central administrative roles in the RATAS project during its different phases.

I am thankful to my wife Ulla for her patient support during all these years. I am also grateful to my children Helena and Mikael for bearing with me during those numerous moments when they have not received my undisturbed attention. I dedicate this thesis to the memory of my late father, who saw the initial phases of the work, and who certainly would have rejoiced in seeing it finally accomplished.

Espoo, November 1995

Bo-Christer Björk

CONTENTS

ABSTRACT **3**

PREFACE **4**

LIST OF PUBLICATIONS **7**

LIST OF ABBREVIATIONS **8**

1 INTRODUCTION **10**

1.1 ROLE OF INFORMATION TECHNOLOGY IN CONSTRUCTION 10
1.2 BARRIERS TO THE INTEGRATED USE OF IT 10
1.3 THE CONCEPT OF COMPUTER-INTEGRATED CONSTRUCTION 12
1.4 INSUFFICIENCY OF CURRENT BUILDING DESCRIPTION METHODS 13
1.5 BUILDING PRODUCT DATA MODELS 14
1.6 STRUCTURE OF THE THESIS 15

2 ORGANISATION, AIMS AND METHODOLOGY OF THE STUDY **16**

2.1 PRECEDING RESEARCH 16
2.2 ORGANISATION OF THE STUDY 16
2.3 AIMS OF THE STUDY 17
2.4 RESEARCH METHODOLOGY 18

3 CONCEPTUAL MODELLING **23**

3.1 THEORETICAL BASIS FOR CONCEPTUAL MODELLING 23
3.2 INFORMATION MODELLING LANGUAGES 26
3.3 TOOLS FOR CONCEPTUAL MODELLING 30
3.4 PRODUCT DATA MODELS AND PRODUCT MODELS 33

4 REVIEW OF THE LITERATURE **35**

4.1 PROBLEM DEFINITION AND EARLY PROTOTYPES (1970-1979) 35
4.2 COMPUTER-AIDED DRAUGHTING AND EXPERT SYSTEMS (1980-1984) 36
4.3 OBJECT-ORIENTATION AND CONCEPTUAL MODELLING (1985-1989) 36
4.3.1 The STEP standardisation effort *36*
4.3.2 Formalised building product data model proposals *37*
4.4 LAYERED PRODUCT DATA MODELS (1990-) 40
4.4.1 A layered architecture of a building product data model *40*
4.4.2 Information modelling language *42*
4.4.3 Generic product data model *43*
4.4.4 Building kernel model *44*
4.4.5 Aspect models *44*

4.4.6 Application models *45*

5 MAIN RESULTS 46

5.1 MAIN RESEARCH ISSUES 46
5.2 REQUIREMENTS FOR BUILDING PRODUCT DATA MODELS 47
5.2.1 Five main requirements *47*
5.2.2 Capability for modelling any information type *48*
5.2.3 Coverage of the information needs of different phases and disciplines *49*
5.2.4 Non-redundancy of information structures *50*
5.2.5 Capability for supporting alternative presentation formats *51*
5.2.6 Independence from restrictions of application software *53*
5.3 INFORMATION STRUCTURES IN BUILDING PRODUCT DATA MODELS 54
5.3.1 Choice of information modelling language *54*
5.3.2 The abstraction hierarchy *55*
5.3.3 The type object mechanism *57*
5.3.4 Modelling spaces and enclosing structures *59*
5.4 INTEGRATION WITH OTHER CONSTRUCTION PROCESS INFORMATION 62

6 DISCUSSION 64

6.1 CONTEXT OF THE ISSUES DEALT WITH 64
6.2 REQUIREMENTS FOR BUILDING PRODUCT DATA MODELS 65
6.2.1 Capability for modelling any information type *65*
6.2.2 Coverage of the information needs of different phases and disciplines *65*
6.2.3 Non-redundancy of information structures *66*
6.2.4 Capability for supporting alternative presentation formats *67*
6.2.5 Independence from restrictions of application software *67*
6.3 INFORMATION STRUCTURES IN BUILDING PRODUCT DATA MODELS 68
6.3.1 Choice of information modelling language *68*
6.3.2 The abstraction hierarchy *69*
6.3.3 The type object mechanism *70*
6.3.4 Modelling spaces and enclosing structures *70*
6.4 INTEGRATION WITH OTHER CONSTRUCTION PROCESS INFORMATION 71

7 CONCLUSIONS 73

7.1 FULFILMENT OF THE AIMS OF THE STUDY 73
7.2 DIRECTIONS FOR FURTHER RESEARCH 74

THE AUTHOR'S CONTRIBUTION TO THE PUBLICATIONS 76

REFERENCES 77

APPENDICES

**Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(http://www.vtt.fi/inf/pdf/)**

LIST OF PUBLICATIONS

I Björk, B.-C. 1989. Basic structure of a proposed building product model. Computer-Aided Design, vol. 21, no. 2, pp. 71-78.

II Björk, B.-C. and Penttilä, H. 1989. A scenario for the development and implementation of a building product model standard. Advances in Engineering Software, vol. 11, no. 4, pp. 176-187.

III Björk, B.-C. and Penttilä, H. 1991. Building product modelling using relational databases, hypermedia software and CAD systems. Microcomputers in Civil Engineering, vol. 6, no. 4, pp. 267-279.

IV Björk, B.-C. 1992. A conceptual model of spaces, space boundaries and enclosing structures. Automation in Construction, vol. 1, no. 3, pp. 193-214.

V Björk, B.-C. 1992. A unified approach for modelling construction information. Building and Environment, vol. 27, no. 2, pp. 173-194.

LIST OF ABBREVIATIONS

CIC
Computer-integrated construction.

COMBINE
Computer Models for the Building Industry in Europe. A European research project financed by the EU through the Joule research programme.

EDIFACT
An international standard for the definition of electronic trade messages.

EDM
Engineering Data Model. An information modelling language designed specifically for product modelling.

EXPRESS
A textual information modelling language which is used for the definition of the international STEP product model standard.

EXPRESS-G
A graphical information modelling language. Implements a subset of the text-based EXPRESS language.

GARM
General Architecture, Engineering and Construction Reference Model. A highly generic building product data model proposal.

IDEF1X
A graphical information modelling language.

IGES
Initial Graphics Exchange Specification. A neutral standardised format for the exchange of geometry data between CAD systems.

NIAM
A graphical information modelling language.

NICK
Neutral intelligent CAD communication. A proposed Swedish standard for the exchange of building product model data.

OOCAD
Object-oriented CAD. A general purpose product modelling environment developed at VTT in the early 1990's.

RATAS
Computer-aided design of buildings. The name for a cluster of research, development and standardisation projects in Finland. Formally organised as a committee under the Building Information Institute.

SADT
A format for defining activity models as hierarchical diagrams.

SfB
A highly influential classification system for construction information. Originally developed in Sweden during the 1950's.

STEP
Standard for the Exchange of Product Model Data. Parts of STEP reached the status of Draft International Standard in 1994 (ISO 10303). The standard is still under development.

UoD
Universe of Discourse. In conceptual modelling theory, the precisely delimited part of the real world which is the subject area for an information modelling effort.

1 INTRODUCTION

1.1 ROLE OF INFORMATION TECHNOLOGY IN CONSTRUCTION

The construction industry is in many respects an information-intensive industry. This is due to many factors: the complexity and size of the end product; the need for visualisation and technical analysis at the design stage; the variety of different know-how and materials needed to erect a building; and the large number of different participants in a construction project. As a consequence, the amount and diversity of information created and referenced during a typical construction project is considerable. The use of information technology (IT) for the processing, management and transfer of data would thus seem to offer great potential benefits to this industry, both in terms of making the design and construction process more efficient and in ensuring better quality of the resulting buildings.

The earliest applications of IT in construction were concerned with technical calculations, an area in which the use of computers is indispensable today. During the 1980's the use of IT for typical construction tasks such as draughting, specifications writing or cost estimation became commonplace, in particular after the advent of the personal computer. The proportion of computer produced documents (of all construction documents) has risen rapidly during the last five years. In some key areas, such as production of architectural drawings, the use of computer support is becoming the rule rather than the exception.

1.2 BARRIERS TO THE INTEGRATED USE OF IT

Despite the tremendous developments of the last few years, the effects of IT on overall information management in construction have not been as significant as might have been expected. In several isolated tasks information is produced more rapidly and efficiently, but the information management and transfer process functions much the same as it did thirty years ago. Today's situation could be described as one of quite extensive computerised document production combined with traditional manual information management techniques.

This state of affairs is partly explained by the poor capabilities of today's applications for utilising each other's data directly in digital form, without a human operator providing the input data. This is true even for the case of fairly homogeneous types of applications, such as CAD-systems from different vendors, and even more so for the transfer of data between heterogeneous applications, for instance between CAD-systems and applications for cost planning or construction management. The problem is accentuated by the fragmented organisational pattern typical of the construction industry.

Technically, the barriers to integration are mainly due to the incompatibility of the fundamental information structures used by different applications for describing a building or aspects of it. In a typical draughting system, a room, for instance, is described only implicitly by the walls surrounding it. In a spreadsheet application describing the client's brief, this room may earlier have been described by one row including cells containing the room's intended usage and required area. A HVAC engineer calculating the building's total heat loss with an analysis program might describe the room's volume as one element in an array of real numbers.

As long as computing applications have only been used for facilitating particular isolated tasks, a situation which by and large prevails today, this incompatibility has had limited consequences. Figure 1 illustrates the "islands of automation" analogy for the pattern by which construction IT has been spreading during the last few decades.

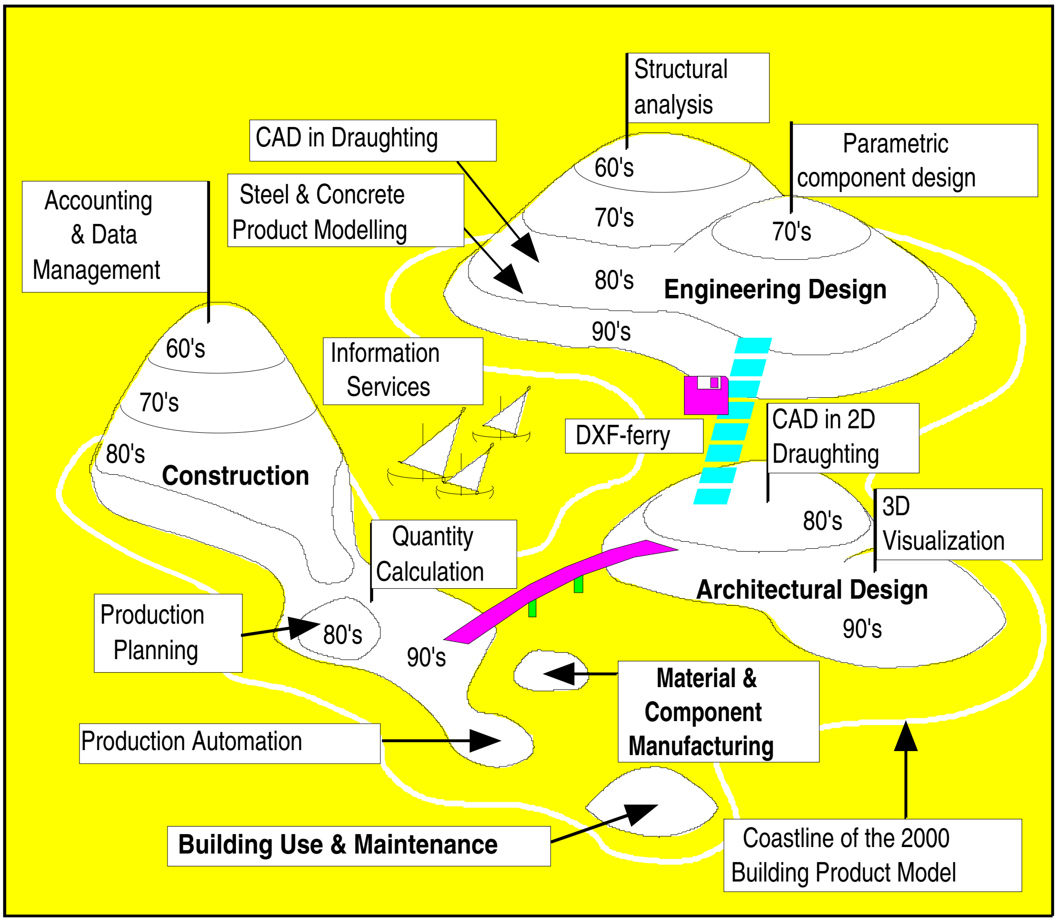

Figure 1. The "islands of automation" analogy is often used to describe the distribution and proliferation of IT use in the construction process. (Courtesy of Matti Hannus and Pär Silén).

But as soon as the organisations participating in construction projects strive to achieve more benefits from IT, these language barriers become a serious obstacle to any development. The need for building bridges between the islands, in the form of standards for the exchange of data, becomes obvious.

The need for integration tools was already recognised by a few leading researchers in the late 1970's [Eastman 1978], [Bijl et al. 1979], and has provided the driving force for a specialised line of research in its own right since about 1985 [NRC 1986], [Christiansson and Karlsson 1990], [Mangin et al. 1991], [Brandon and Betts 1995]. Construction professionals involved in IT development are also increasingly becoming aware of this need. As a consequence, a market demand for techniques and software for such integration is emerging.

1.3 THE CONCEPT OF COMPUTER-INTEGRATED CONSTRUCTION

Many researchers have recently adopted the term computer-integrated construction (CIC) to describe a visionary target state for the use of IT in construction. The term has been derived from the computer-integrated manufacturing (CIM) concept, which is used in other branches of industry. CIC is seen as one of the few means the construction industry has of increasing its productivity and the quality of its end product [Saarnivaara 1990], [Dupagne 1991].

Computer-integrated construction is characterised by both the use of computing for all kinds of applications and by the integration of these applications via data transfer networks and transfer standards [Wright 1988], [Howard et al. 1989]. Functionally computer-integrated construction can be defined as a construction process characterised by very extensive computer-aided integration of all information processing tasks, whether these are performed by humans or computers. In the context of CIC, integration can primarily be understood to mean efficient information sharing and data exchange, using IT as the enabling technology. All the information which is required in a particular task should be accessible almost instantly in a form which can be manipulated using computers.

The definition of the scope of CIC differs from one author to another. The large Japanese construction firms represent a very comprehensive approach, in which information processing and management, construction systems (prefabricated components and assembly methods) and on-site robotics systems are developed as an integrated total system [Yamazaki 1990]. This approach seems particularly suited to very large construction companies which carry out both the design and the construction of the projects they are involved in. It seems less adequate for the more open type of construction process typical of the European market. A more narrow definition of CIC would emphasise the information management and transfer aspects in an industry where the partners in projects constantly change.

Some of the key ingredients of CIC were identified in a research project carried out by the Technical Research Centre of Finland [Björk et al. 1991a]. They include an infrastructure of computer networks, availability of public and commercial databases for general construction information, and agreements on data transfer standards. All of these ingredients are necessary prerequisites of CIC, but not as such sufficient conditions for reaching that stage.

Some of these ingredients will be provided as a result of general developments in IT technology. Others are dependent on explicit actions initiated from within the construction industry. As a consequence, IT strategies are important both on the level of the individual firm and of the construction industry as a whole [Betts 1992]. Today, some pioneering firms in various countries are making CIC an important part of their overall business strategies [Laitinen 1992], [Miyatake et al. 1992]. On the industry level, ambitious R&D strategies have been initiated in a number of countries such as Norway [NTNF 1984], Finland, Singapore [Betts et al. 1989], Denmark [ATV 1994], and the UK [BT 1995],. The Finnish RATAS activities, which have provided an environment for the research described in this thesis, offer a good example of such an initiative [Björk 1994b]. The development of building product model technology, the subject area of this thesis, has been recognised as a central feature of the Finnish strategy since 1988.

1.4 INSUFFICIENCY OF CURRENT BUILDING DESCRIPTION METHODS

The methods for describing buildings have evolved over many centuries, ranging from textual descriptions and miniature models to sophisticated computer-generated visualisations. During the last couple of centuries, drawings on paper have been the dominant method.

The documentation methods in use today have been shaped by the needs of designers to transfer their design decisions to each other and to construction companies in an unambiguous form, as well as by the limitations of the available technology (paper and pen). Information technology has so far been used mainly as a means of facilitating the production of traditional types of documents and has had limited impact on the presentation formats and information content of such documents.

As a consequence of the above, a typical building description, whether manually or digitally produced, has until recently consisted of a set of loosely interrelated documents. (More integrated 3-D models have been used for special purposes such as visualisation, but not usually for detailed design and production documentation). A large part of the information is redundantly represented in many different documents. Human experts are needed to interprete the information described in document header information and in the documents themselves in order to integrate the information from separate documents in a meaningful way. Some help is provided by classification systems for building elements, which can assist in interrelating particular drawing entities with specification clauses and bill of quantity items. Building element table standards have been defined in many countries, in addition to standards for the format and content of different types of drawings. The use of so-called reference files in commercial CAD software [for a good description cf. Abb 1993], is also facilitating a move towards the use of more comprehensive integrated building models.

1.5 BUILDING PRODUCT DATA MODELS

Recent developments in database technology combined with the rapidly increasing processing power of computers, now make it possible to store all the information contained in traditional building documents in a centralised or distributed database as an integrated, comprehensive building model. The structure of such a model, for which the term building product model is commonly used, has been the subject of quite intensive research for the last few years.

In a concise form the overall functional goal of building product model research can be stated as follows:

How should we structure digital information describing a building in order to facilitate as much as possible, the exchange of information between the computing applications in construction which produce or utilise this data.

The research effort described in this thesis has dealt with a number of information structures needed to define how information is stored in such building product models.

Information management based on product models would change the techniques of design documentation quite radically. Figure 2 illustrates the differences between the document-oriented and the product model-oriented approach to organising design information. In the product model approach, documents resembling traditional drawings and specifications would only exist as user views of the information contained in databases. Computer applications needing information would query the product models directly rather than rely on human operators interpreting documents. The redundancy and updating problems typical for current documentation methods would be easier to solve. New, user-friendly types of documentation (i.e. hypermedia interfaces) could be developed.

Product models can, of course, be equally useful in any industry involving the design and manufacturing of physical artefacts (i.e. airplanes, cars, ships, machines). Research concerning object-oriented product models has been going on actively since about 1985. The international STEP standardisation work, an ISO activity for defining a standard for the representation and exchange of product model data, was initiated in 1985 and is still continuing [ISO 1992a]. STEP involves the participation of hundreds of experts from several countries and is indirectly supported by large national and international research projects.

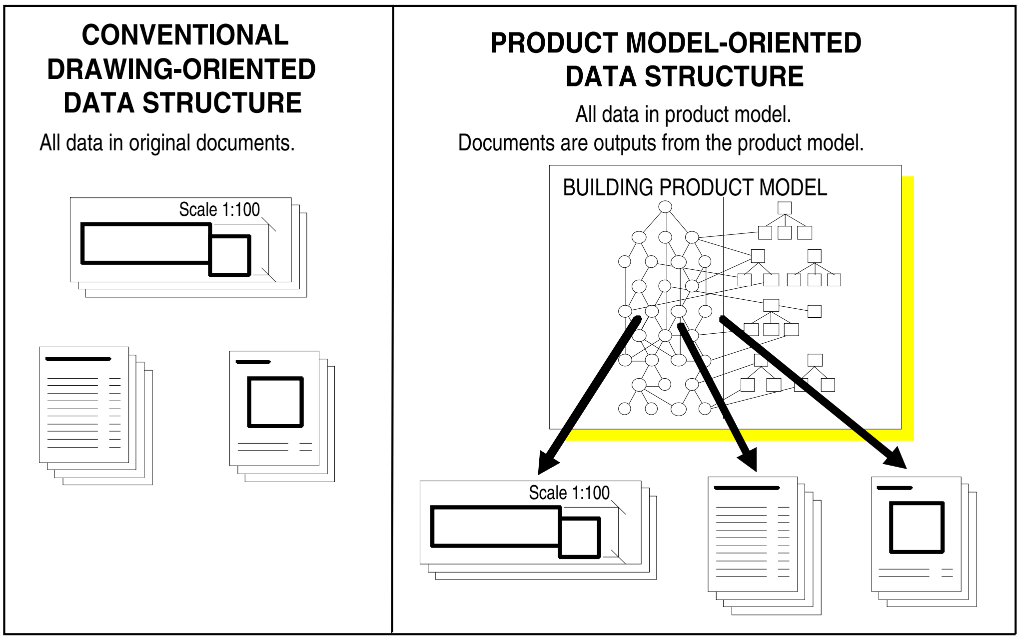

Figure 2. The principles of data organisation in product models differ radically from the document-centred approach of the traditional design process.

1.6 STRUCTURE OF THE THESIS

This thesis consists of five articles, which have been published in scientific journals during 1989-92, supplemented by this summarising report. In order to make the report as up-to-date as possible, some proposals and results of other researchers, which have been published after completion of the above-mentioned five articles, are included in the state of the art and discussion sections. As a consequence, this author's past results are to some extent reinterpreted and evaluated in the light of later developments.

This report is organised as follows. The opening section discusses the problems of information management in construction and the concept of computer integrated construction. This should be a sufficient background for understanding the basic aims of building product model research. The second section explains the background, organisation, aims and methodology of the research effort which resulted in this thesis. The third section is a short review of the fundamental concepts of conceptual modelling and introduces most of the concepts which are necessary to understand the technical content of the rest of the report. Section 4 is a review of relevant research and the current state of the art in this field. In particular it discusses a framework for building product modelling which uses a kernel model supplemented by aspect models. This framework is in section 5 used to position the research results of this thesis. Section 6 contains a critical review of the results, comparing them with other relevant work. The last section compares the results of the study with the original aims and proposes some topics for further research.

2 ORGANISATION, AIMS AND METHODOLOGY OF THE STUDY

2.1 PRECEDING RESEARCH

As a researcher at the Technical Research Centre of Finland (VTT), the author has been involved in research concerning computers in construction since 1982. He was the principal investigator in a project entitled *Integrated computer-aided building design,* which involved several experts from the industry. In the project's final report, a number of areas for standardisation which were considered important for the future integration of computer applications in construction, were listed [Björk and Keppo 1984]. The above-mentioned project was one of the driving forces behind the founding of the RACAD association [Björk 1987], which originally initiated the research and standardisation program known by its acronym RATAS.

RATAS activities started with a prestudy in 1985 [Sarja and Leppänen 1987]. During the second phase, work was carried out in four working groups, dealing with general construction information services, data transfer standards, a standard for the representation of a building as a data base and new documentation standards [Enkovaara et al. 1988]. The proposal for an overall organisation of a building product model made by the third working group is quite well-known abroad as the RATAS model [Hannus 1987], [Björk 1988], [RATAS-Committee 1988] .

2.2 ORGANISATION OF THE STUDY

The RATAS working group had no resources in 1988 for further work on the product model proposal and there was an urgent need to deepen the theoretical foundations related to the subject. For this reason VTT decided, as part of a larger research programme concerning *Information and Automation Systems in Construction*, to launch a three-year research project for further study of the building product model approach. This project is hereafter referred to as *VTT's product model project*.

The author had overall responsibility for planning and execution of the project. The major part of the work reported in this thesis was performed within the above-mentioned project during the years 1988-91.

In addition to results from VTT's product model project, the thesis also contains material that resulted from the author's participation in a number of later spin-off projects during 1989-93.

These projects included:

- a prestudy concerning the use of the product model approach for energy-conscious design of buildings [Talonpoika et al. 1990],
- a subtask of the third phase of the RATAS project, dealing with the application of the product model approach to quantity take-off [Penttilä et al. 1991],
- a study concerning overall strategies for Computer Integrated Construction [Björk et al. 1991a],
- the European Union-funded COMBINE project [Augenbroe 1992].

Despite the fact that the work reported in this thesis has been carried out in several organisationally separate projects, for the sake of simplicity it will subsequently be referred to in the text as one coherent study.

2.3 AIMS OF THE STUDY

The main thrust of the study was not to find out whether an object-oriented product model description of a building as such, is better suited for integrating the computer applications in construction than traditional documentation produced by computer software. The quickly growing research literature in this area bears witness to the fact that such a paradigm is already shared by part of the scientific community. The awareness among leading construction companies and software developers of the potential of this approach, is also growing [Kita et al. 1990], [Beuke and Ranglack 1993].

Instead, the focus of the study has been to investigate ways of structuring information in object-oriented building product models. The starting point for the study was the framework model proposed by the RATAS project in 1987.

Seen as a whole, the aims of this study coincide to a large extent with those which were originally defined for VTT's product model project in 1988. Some of these aims were later included in the objectives of the spin-off projects mentioned above.

The aims of the product model project were:

- To elaborate on the criteria which a building product data model should fulfil and to study the fulfilment of these criteria using prototypes.

- To study in detail a number of critical generic information structures which are needed in building product data models.

- To test the applicability of the approach by building prototypes using a variety of different software techniques.

- To develop innovative user interfaces to design information structured according to the product model approach.

- To model a whole building in detail as a test case.

- To demonstrate the exchange of data between building product models on the one hand, and technical analysis software and expert systems on the other.

- To study how the information included in building product data models can be integrated with other information used in construction projects (i.e. information describing activities, resources).

The fulfilment of these aims is discussed after the presentation of the results in section 7.1.

During its short history this research domain, which to an outside observer might seem very narrow, has turned out to be multi-faceted and extensive. Consequently, this study had to be restricted to only some of the many important issues which need to be resolved for product modelling technology to become commercially implementable. Topics excluded from this study include physical transfer and storage of product model data, modelling of design versions, data integrity management and modelling of shape and location information. The majority of these are generic product modelling topics common to most branches of industry and considerable efforts are currently being made world-wide to find appropriate solutions.

2.4 RESEARCH METHODOLOGY

Design theory is a natural starting point for any discussion of product modelling as an academic discipline [for a good overview of the evolution of design theory cf. Lundequist 1992]. Design theory as an independent discipline evolved during the 1960's [Alexander 1964], [Simon 1981], strongly influenced by developments in operations analysis, systems theory, cognitive science and management studies. Since the start, the rapid developments in computing technology have had a strong impact on the evolution of design theory.

The use of computers to support the design process can be roughly divided into two separate domains or aspects. The first domain consists of the formal modelling of *problem solving processes* in design [Gero 1994]. A central question which many researchers have asked is how far the design process can be automated using computer support [Kalay 1987]. During the 60's and 70's considerable efforts were, for instance, devoted to the definition of algorithms for the automated synthesis of spatial layouts. These efforts were strongly influenced by developments within operations analysis. In later years, the use of knowledge-based systems for the automation of design decision-making has been an important research direction. Knowledge-based tools have so far, however, found very limited use in day-to-day design practice.

The second domain consists of the *representation of buildings* in digital form. In this domain, the development of commercial CAD-systems has had a profound effect on design practice. Although the acronym CAD means computer-aided design, indicating that such systems also include support for the problems solving process, the overwhelming majority of systems are either draughting or geometric modelling tools. Since the 1970's, researchers have studied alternative building representation methods (building product models) using techniques borrowed from database and programming language theory.

The current strong research interest in building product models can be interpreted as a typical example of paradigm shift, in line with the theory of the development of scientific disciplines originally put forward by Kuhn [the reference used here is Molander 1990]. This paradigm shift seems to consists of two highly interrelated elements:

- A paradigm shift in computer software techniques from procedural algorithmic programming languages to object-orientation.

- A paradigm shift in the representation of buildings from drawings and written specifications to integrated models taking building components and spaces as their starting point.

For at least a century, the representation of buildings as a set of drawings supplemented by written specifications has been the prevailing paradigm. This technique has been taught in architectural and engineering schools, and has been embedded in industry practice rules, standard symbols etc. The first wave of computerisation (computer-aided draughting) has only helped in providing support for more efficient production of the same documents as before. Building product modelling represents a fundamentally different way of describing buildings.

As a research domain, the study of building product models is still relatively young. Despite early pioneering work in the late 1970's, it started to emerge as a recognised field of research only around 1987-88 [Christiansson and Karlsson 1990]. In the research work published so far, the emphasis has been on presenting results in the form of conceptual schemas and prototype implementations. The discussions on methodological issues have centred on arguments in favour of using particular information modelling techniques, and not so much on discussing the process of defining conceptual schemas.

A survey of the literature (for references to the relevant projects see sections 4.3 and 4.4 below) reveals that the research strategy which de facto has been followed in most research projects consists of a number of stages:

- Choice of a particular information modelling methodology.
- Gathering of empirical knowledge about the building information needed in the different phases of the design and construction process.
- Definition of information structures as more or less formalised conceptual schemas.
- Experiments with prototypes to test the conceptual schemas.

The first stage consists of the choice of a particular information modelling methodology or language to be used for constructing the conceptual schemas. There are very few examples where the presented schemas have been defined in parallel using several modelling languages. One small example is described by [Eastman and Fereshetian 1994] who modelled the same schema using five different languages in order to compare the capabilities of these.

The second stage consists of gathering empirical knowledge about building information needed in the different phases of the design and construction process. This knowledge can concern the needs of human information users, in which case a study of existing building documentation methods can be carried out. Alternatively, a study can be made of the input and output of existing computer applications. In the COMBINE project, for example, the data manipulated by six different building design and analysis applications were studied in detail as a basis for the definition of the integrating conceptual schema [Dubois et al. 1992].

Sometimes this information gathering is done in a highly structured way using data flow diagrams, data tables, activity diagrams etc. In other projects, in particular if only highly generic data structures are defined, the participating researchers rely on their general knowledge of building technology and design methodology.

The third stage consists of the definition of information structures as more or less formalised conceptual schemas. There are numerous research reports, conference papers and articles in scientific journals which present proposed schemas. On the other hand, the process of arriving at such schemas (criteria and procedures for choice of object classes etc.) has received less attention. Defining a conceptual schema seems to be more like a craft, to be compared with architectural design, than scientific analysis.

The fourth stage consists of experiments with prototypes to test the conceptual schemas. Such prototype work can be subdivided into two major categories, based on the software tools used. The first option is to develop the prototypes from scratch using basic programming languages, possibly enhanced by software tools for building graphical user interfaces. The major benefit of this option is that it poses few restrictions on the freedom of researchers to implement the conceptual schemas which have resulted from the theoretical work. The major drawback is that it is very labour intensive in terms of programming effort.

The second option is to base the prototype development on some existing general purpose CAD systems, possibly in combination with other types of software such as relational databases. If this option is chosen, the applications used may quite strongly influence the conceptual schemas which can be implemented. On the other hand, a large part of the functionality and user interfaces of the prototypes is achieved with little effort. It may also be easier to argue for the validity of the product model approach in discussions with practitioners, if well-known tools are used and the user interfaces are of a high quality.

It is difficult to draw very definitive conclusions on the usefulness of the conceptual schemas proposed from the reported successful development of prototypes. This is due to two main reasons. Firstly, the majority of these prototypes have only handled very small subsets of all the building description data which in real design projects should be managed using building product models. Full-scale tests, either as dry runs in laboratory conditions or in real design and construction situations have not been reported so far. Secondly, there are no reports, to this author's knowledge, of prototypes having been tested outside the research laboratories by independent third parties, who have not participated in some way in the definition of the conceptual schemas implemented in the prototypes.

In the case of failures in the development of prototypes, they do not necessarily rule out the usefulness of the information structures proposed, since the problems may have been due to the particular hardware and software tools used in the prototype work, lack of resources etc. An additional problem with the prototype experiments reported so far is that they usually are impossible to replicate, due to the heterogeneity of the test material and the software applications used for the prototype work. From this viewpoint, it could be useful if a specific, well-documented building design could be used as a test case for comparing the performance of different building product data model proposals and/or prototypes.

The above process with four stages is a description of what is actually done in most research projects in this domain. A very useful fifth stage would consist of the testing and empirical study of product model based systems in real design work. This would, for instance, enable researchers to study how product model based systems could change the design process itself. Unfortunately, very few of the research prototypes that have been produced, have reached a stage of maturity for testing in real design projects. Despite a growing interest in this technique among software developers, there are very few systems in general use which could be termed product model based.

The research strategy of this study tried to take into account the issues discussed above and to benefit from the experiences of earlier and parallel projects. The study was carried out as a combination of theoretical analysis and prototype development, in line with the methodology used by the majority of other researchers. The theoretical work was carried out based partly on the knowledge of the involved researchers, and partly on feedback from a number of industry experts. In the earlier phases, the theoretical work resulted in a number of proposed overall principles for building product modelling and produced examples of information structured according to these principles. Important parts of the work concerned strategies for the development of building product data models. In the later stages, information modelling techniques were also utilised for formalising conceptual schemas.

For the experimental part, a strategy of small incremental prototypes was adopted. Rather than trying to develop one large prototype which could handle information on the scale of a whole building, as originally envisaged, the researchers used small subsets of all the information used for describing a full building and tried to tackle specific and limited problems with each prototype. Commercially available CAD systems, relational databases and hypermedia software were used for the development work, in order to minimise the time spent on programming issues and to maximise the time spent on solving the problems at hand.

3 CONCEPTUAL MODELLING

3.1 THEORETICAL BASIS FOR CONCEPTUAL MODELLING

The foundation for the development of product models is provided by the theory of conceptual modelling [Brodie et al. 1984], [ISO 1985], [Boman et al. 1991], a branch of database research which has evolved during the last twenty years. Conceptual modelling is typically the first stage in the process of developing database applications, and its use in the development of computer applications in general is increasing. There are, for instance, several so-called object oriented analysis and design methods to choose from [Coad and Yourdon 1991], [Rumbaugh et al. 1991], [Booch 1994].

In conceptual modelling, a basic distinction is made between those phenomena of the real world which we are interested in (in our case buildings) and the concepts which we use for describing them in information processing systems. The real world is perceived as consisting of objects or entities. Objects have properties, and they may be interrelated. Objects which to some degree have similar properties, can be grouped into categories or classes.

In conceptual modelling theory, the set of real world objects which constitutes the subject area for some particular information modelling task, is called the universe of discourse (abbr. UoD). Often the UoD is defined to include all possible states of the real world objects as they evolve over time. A typical example where the UoD has this temporal dimension is a database system for keeping track of the students enrolled in a university, the exams they pass and the degrees they are awarded.

In information processing systems, the objects of the real world are represented by linguistic symbols. For instance, the symbol or term *The White House* in the English language refers to a particular building in Washington D. C. There are two basic categories of such symbols. Some symbols represent individual objects in the real world, other symbols represent whole classes of objects. The latter category of symbols are often called types [Boman et al. 1991]. Thus *The white house* is a term symbolising an individual real object and *building* is a term corresponding to the class of real buildings.

The objects of the UoD have properties. In the information processing world such properties are represented by attributes. Attributes can be used both to represent simple properties (i.e. the height of a person) and associations between separate objects (i.e. Bill Clinton being married to Hillary Clinton).

Logical propositions which state something about the properties of objects within some UoD, can be either true or false. The proposition that Bill Clinton is married to Hillary Clinton, for instance, is true. The proposition that the White House is situated in New York is false.

There are two fundamental categories of propositions related to some particular UoD. The first ones are propositions that hold for all possible states of the UoD over time; necessary propositions. They thus define which objects may and in some cases must occur in the UoD as well as the rules and constraints which apply to them. The second category of propositions are true only for particular objects at specific points in time. These can be called non-necessary propositions. A necessary proposition for the UoD of all human beings is that each person has exactly one biological mother and father. That King Karl Gustav and Queen Silvia are the parents of Princess Victoria is an example of a non-necessary proposition.

Consider the total set of propositions which are true about a particular state of the UoD at some point in time. Based on the nature of the propositions, we can group them into two separate sets; a *conceptual schema* and an *information base*. The set of necessary propositions constitutes the conceptual schema and the set of non-necessary propositions an information base. The non-necessary propositions of the information base cannot contradict the necessary propositions. An information base can thus be regarded as a sort of "snapshot" of the universe of discourse at some state in time, whereas the conceptual schema formalises the laws which govern the UoD in general. Several different information bases can conform to the same conceptual schema.

A conceptual schema thus specifies which types and attributes can be included in a valid description of the UoD. In addition, a conceptual schema usually includes different kinds of rules and constraints which govern the possible states of the information bases conforming to the schema. Static rules establish dependencies between parts of a system at any given instant of time. Dynamic rules are those which govern the evolution of the system over time. Derivation rules specify how some attributes can be derived from other attributes.

The presentation above is rather abstract. On a more practical level the conceptual schema corresponds quite closely to the data declarations of a particular data base application, or the class definitions included in a program written in some object oriented programming language. The information base corresponds to the data stored in the database at some instance in time, or to the data manipulated at run time by the object-oriented application. The use of the term information base in contrast to database, emphasises the interest in the logical content of the information rather than its physical storage in records and tables.

A well-known framework for discussing the role of conceptual schemas in data base systems is the ANSI-SPARC three layer architecture [the reference used here is ISO 1985]. This architecture distinguishes between *conceptual, internal and external* schemas or views. Conceptual schemas have been discussed above. An internal schema is a schema defining how the information defined in a conceptual schema is represented internally within some particular application environment, for instance in a particular database system. External schemas correspond to the way particular end users look selectively at the information (i.e. the structural designer's view of a building). There can be several internal and external schemas corresponding to each conceptual schema.

The most fundamental way of representing the information contained in conceptual schemas and information bases is by using formal logic. For this reason, logic programming languages, such as Prolog, have been proposed as suitable languages for conceptual modelling [Boman et al. 1991]. An approach based on logic programming could also be useful as a neutral intermediate format for the integration of conceptual schemas formulated in different information modelling languages [Johannesson 1993], [Fulton 1995].

As an illustrative example of using formal logic for representing an information base, table 1 contains a number of propositions about the White House formulated in a Prolog-like syntax [for a description of Prolog cf. Sterling and Shapiro 1986].

Table 1. Some propositions about the White House formulated in a Prolog-like syntax.

Proposition	Explanation
Building (White House)	The White House is a building
Room (Oval Office)	The Oval office is a room
Window (W-347)	The object W-347 is a window
Located in (Oval office, White house)	The Oval office is in the white house
Window of (W-347, Oval office)	W-347 is a window of the oval office

The propositions in the table are non-necessary propositions and belong to the information base describing the White House. The one-place propositions are type statements, stating to which types certain objects belong, whereas the two-place propositions are attribute statements representing associations between objects.

However, this kind of technique is rarely used in practice for the development of applications. Software technology provides a large variety of mechanisms for grouping such "atomic" propositions into larger structures or "molecules", which improves both programming and computing efficiency. Using such mechanisms, the individual propositions of the conceptual schema and the information base can be grouped into higher level aggregates. Examples of modelling approaches based on grouping can be found both in programming languages (i.e. the objects of object-oriented programming), database systems (entity types) and knowledge-based systems (frames). We will look at some basic mechanisms used for this purpose in the following section.

3.2 INFORMATION MODELLING LANGUAGES

On a generic level, different terms have been used to denote languages for the definition of conceptual schemas. These *include information modelling language*, *conceptual modelling language*, *conceptual modelling approach*, *data model* [Brodie et al. 1985], *general-purpose data model* [Eastman and Fereshetian 1994] and *minimal conceptual schema* [ISO 1985]. The term *data model* is unfortunately often also used as a synonym for conceptual schema, which may lead to some confusion.

It is important to bear in mind the varying emphasis placed by different approaches on modelling the logic of the UoD as opposed to structuring data efficiently in computer memory. Often the term *information* is used if the emphasis is predominantly on modelling the logic of the UoD, and the term *data* if more emphasis is placed on how this information is efficiently stored and manipulated in files and databases. The modelling of integrity constraints (for instance, the necessary proposition that each person has exactly one mother and one father) is a typical information modelling issue. The encapsulation or data hiding principle in object oriented programming, or the normalisation techniques of relational databases [Date 1987] are typical features related to data structuring and programming efficiency.

Since the emphasis in this study is on developing conceptual schemas which are independent from particular implementation techniques, the term *information modelling language* will primarily be used in the following.

There are a number of fundamental mechanisms which can be found in most information modelling languages. These include:

- Type declarations for grouping all attributes, integrity rules etc. related to some particular class of objects (in the UoD sense).
- Attributes whose values are primitive or composite types.
- Inbuilt mechanisms for defining certain standard types of integrity constraints.
- General facilities for defining more complicated rules and constraints.
- Inheritance mechanisms.

In the following, these mechanisms are illustrated with a simple example, formulated in the EXPRESS information modelling language [ISO 1992c]. The choice of EXPRESS for this purpose is based on the author's familiarity with the language as well is its current widespread use among building product model researchers. The example is purely illustrative, and should not be considered part of the modelling efforts presented later on in this report. The use of bold characters in the examples serves only for highlighting the features of interest and has no significance in the schemas as such.

Type declarations are usually called object classes or entity types (In EXPRESS simply entities). The declarations typically begin by defining a name for the class, followed by a list of attribute definitions applying to it. Instances of such classes will be created in databases or as temporary data manipulated by applications. An instance represents some individual object in the UoD, and fills the template data structures of the class with the values for that particular object. Often such instances are called objects or entities (as opposed to classes). The terminology can be very confusing since the same terms are used to mean both concepts from the UoD and from the conceptual world, depending on the modelling approach. Usually the context helps in explaining which usage is intended.

The following EXPRESS code defines some attributes needed for describing spaces in a building.

```
ENTITY space;
space_id : STRING;
area : REAL;
END_ENTITY;
```

In more complex information modelling languages the data types of attributes can be either primitive or composite. A primitive type is a basic data type such as an integer, a real number, a string of characters or a truth-value {false, true}. A primitive type cannot be broken down into further parts. Both attributes in the definition of the *space* entity are primitive types.

```
ENTITY space;
space_id : STRING;
area : REAL;
END_ENTITY;
```

A composite type is constructed from a number of primitive types by some well-defined operations. A good example of a composite type is the use of the *space* class as the data type of the attribute *encloses* in the *wall* class definition shown below.

ENTITY wall
includes_window : SET [0:?] OF **window**;
includes_door : SET [0:?] OF **door**;
encloses : SET [1:?] OF **space**;
END_ENTITY;

Integrity constraints restrict the permissible values of attributes. They can, for instance, specify how many instances can be found on each side of an attribute (cardinality). An example of this kind of constraint is that a wall includes zero-to-many windows and encloses one-to-many spaces.

ENTITY wall
includes_window : **SET [0:?]** OF window;
includes_door : **SET [0:?]** OF door;
encloses : **SET [1:?]** OF space;
END_ENTITY;

Another important type of constraint specifies that certain instances are dependent on the existence of other instances (referential integrity). Information modelling languages usually contain specific syntax for modelling of the most common generic constraint types. The NIAM language, for instance, contains quite powerful mechanisms for this purpose [Nijssen and Halpin 1989].

Many modelling languages contain methods for defining more complicated rules and constraints involving arithmetic and boolean operations. Such methods can be used in particular for the definition of derivation rules and rules involving objects from several different classes. In the following, we have added two more attributes (height and volume) to the definition of the space class, as well as some rules restricting their values.

ENTITY space;
space_id: STRING;
area : REAL;
height : REAL:
DERIVE
volume : REAL := area * height;
WHERE
WR1 : area > 0.0
WR2 : height > 0.0
END_ENTITY;

Inheritance is a feature which can be extremely useful in complicated modelling tasks. It forms the backbone of the object-oriented programming paradigm. The terms generalisation and specialisation are also often used to describe the inheritance mechanism. Classes defined as subclasses inherit the attributes and constraints of their superclasses. There are some variations on inheritance, for instance single versus multiple inheritance.

The following EXPRESS code shows how the classes for describing windows, doors and walls inherit the attribute for a unique identifier from the superclass building part.

```
ENTITY building_part
SUPERTYPE OF (window, door, wall);
id : STRING;
UNIQUE
UR1: id;
END_ENTITY;

ENTITY wall
SUBTYPE OF (building_part);
includes_window : SET [0:?] OF window;
includes_door : SET [0:?] OF door;
encloses : SET [1:?] OF space;
END_ENTITY;

ENTITY window
SUBTYPE OF (building_part);
serves_space : space;
END_ENTITY;

ENTITY door
SUBTYPE OF (building_part);
serves_space : SET [1:2] OF space;
END_ENTITY;
```

A detailed discussion of the structures and merits of different information modelling languages is beyond the scope of this text. The ISO technical report on *the concepts and terminology for the conceptual schema and the information base*, for instance, distinguishes between the following generic categories of modelling approaches [ISO 1985, p. 69]:

- Entity-attribute-relationship approaches
- Binary and elementary N-ary relationship approaches
- Interpreted predicate logic approaches

One information structuring mechanism which may cause some confusion is the relationship between entities. Some approaches, in particular the entity-relationship approach [Chen 1976], model relationships explicitly as a category different from attributes. Other approaches use attributes where the data type of the attribute is another entity for achieving more or less the same purpose.

When modelling artefacts consisting of physical components, it would seem intuitively appealing to make a clear distinction between attributes and relationships. The building components themselves would be modelled by object classes (i.e. *wall*, *window*). Functional relationships between such components would be modelled as relationships (i.e. *serves*, *encloses*) and what we conceive as properties of singular physical objects (i.e. *height*, *area*) as attributes. The use of certain modelling languages almost automatically leads to this (i.e. IDEF1X), whereas the use of other languages makes it less evident (i.e. NIAM, EXPRESS). Even if the information modelling language in question makes a clear separation between entities, relationships and attributes, it can often be difficult to decide which structure to use when modelling a particular information item.

A term which will be used frequently in this report is *information structure*. The definition is non-scientific, but can be useful for readers because of the frequent use of the term.

For the purpose of this report *information structure* is defined to refer to any grouping of logical propositions, used as a predefined template. Information structures can exist at several levels of abstraction. Examples of information structures could at the basic information modelling level be the object class, the frame etc. Thus the fact that an information modelling language makes a separation between attributes and relationships could be seen as an issue of information structure. On a higher level of abstraction, information structures can thus be seen as the "substance", the network of information items and links between them, from which conceptual schemas are built. Both single object classes and configurations of several classes have information structures. Full conceptual schemas are more complete than individual information structure in the sense of modelling all the information needed to describe some UoD.

Another term frequently used in the literature on database and programming language theory is *data structure* [cf. Louden 1993, p.6]. The term data structure stresses more physical implementation in computer files and as variables in programming languages. Information structure is more concerned with the conceptual level of how to model the UoD.

3.3 TOOLS FOR CONCEPTUAL MODELLING

The information modelling languages which have evolved fall into two major categories, graphical diagram techniques and textual definition languages.

Graphical diagram techniques are especially useful in the early modelling phases. They are also good for presenting conceptual schemas to other researchers or application developers. As the number of classes grows into the hundreds, the diagrams become difficult to handle and update, especially if they have been drawn with general purpose draughting software. Examples of graphical techniques which have been used for the definition of product data models include IDEF1X [Appleton 1985], NIAM [Nijssen and Halpin 1989] and EXPRESS-G.

Textual definition languages are better suited for computer processing and for handling large models. Such languages also offer better possibilities for the definition of rules and methods, which are difficult to show with simple graphical symbols. A good example is the full EXPRESS language [Schenck and Wilson 1994], which is increasingly popular also outside its originally intended use for the definition of the STEP product model standard.

Figure 3 shows the example introduced earlier in section 3.2, modelled using both the graphical and the textual versions of the EXPRESS language, as well as an example of objects instantiated from this schema. The reader is cautioned that the model contains some simplifications. The location and geometry information needed for generating the drawing have, for instance, been omitted.

The software technology supporting conceptual modelling has evolved quickly during the last few years, for instance as a part of so-called CASE tools. For EXPRESS in particular, there are now several categories of software which can be used for the development and management of models [Wilson 1992], [Karstila 1993], [Poyet et al. 1993]. These include:

- Software for drawing graphical conceptual schemas.
- Hypertext browsers for textual conceptual schemas.
- Parsers which check the syntax of text schemas.
- Tools which can convert a conceptual schema from NIAM to EXPRESS.
- Generators which, from an EXPRESS schema, can create code in object-oriented programming languages or table definition statements for relational databases.

Similar kinds of software tools can be found for other information modelling languages. The trend seems to be towards handling a conceptual schema in the same way as design information in a product model. This means that draughting and hypertext systems can be used as user interfaces to the same underlying conceptual schema, not merely to create separate documents which can be related only via human interpretation.

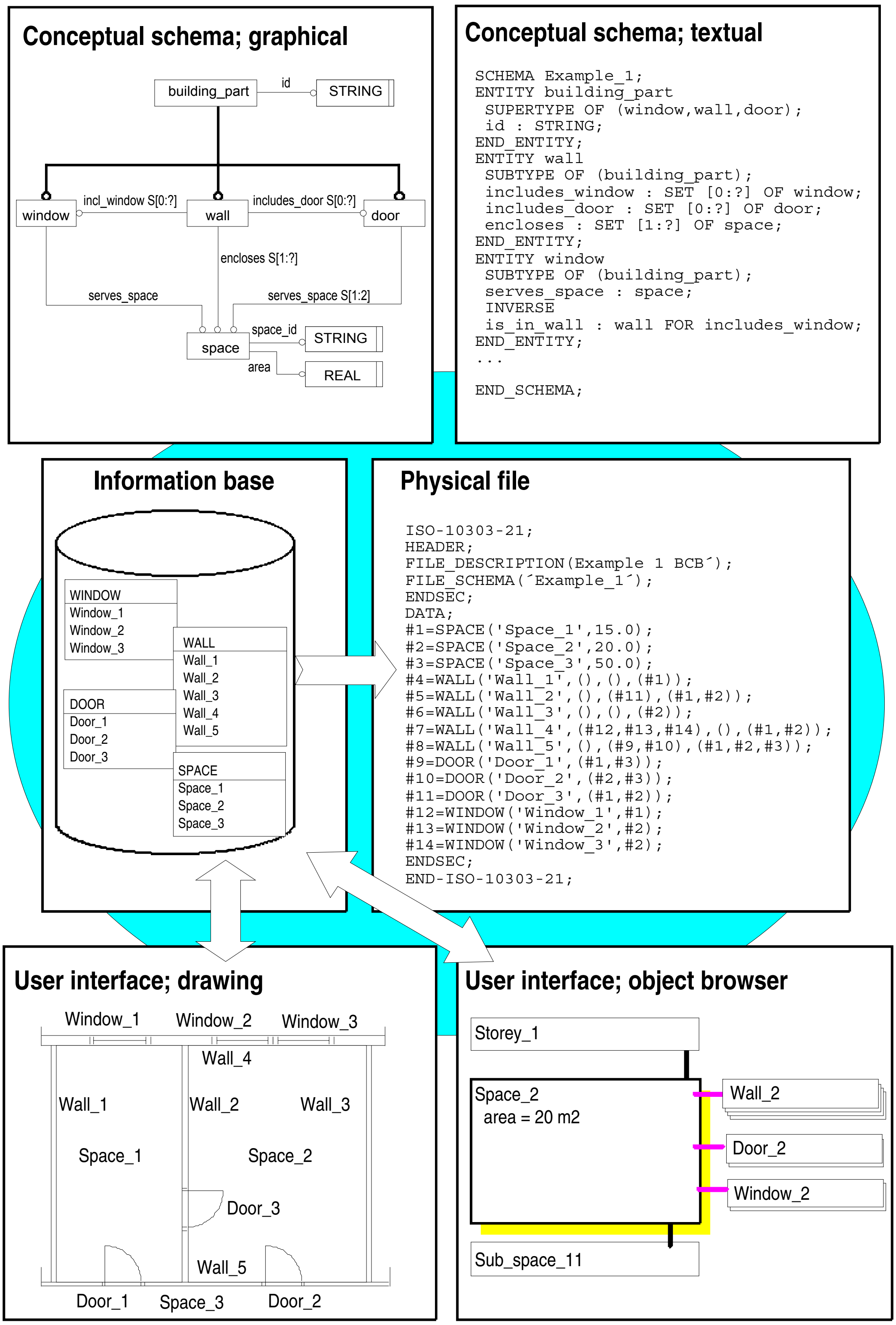

Figure 3. An example of the definition of a small conceptual schema in both EXPRESS-G and EXPRESS. The middle part of the figure illustrates how instances of this schema would be stored in a database or transferred as a STEP physical file. The lower part shows how a human user could view the information as a drawing or in an object browser.

3.4 PRODUCT DATA MODELS AND PRODUCT MODELS

The purpose of sections 3.1-3.3 has been twofold. Firstly, to familiarise readers who are unfamiliar with conceptual modelling theory with some basic concepts. Secondly, to enable us to make precise definitions of some key terms of product modelling.

A product data model is a particular type of conceptual schema, expressed in some information modelling language. The feature which distinguishes a product data model from other conceptual schemas is that the universe of discourse consists of some type of artefact designed and manufactured by man. In addition, a product data model describes the component objects of the artefact in question directly, and not indirectly by modelling the information content of documents describing the artefact, such as drawings. The UoD should cover the whole life-time of the artefact, not only the design or manufacturing stage.

In the literature on the subject, the terms *product data model* and *product model* are both used to denote this particular type of conceptual schema, which can seem confusing. This author is of the opinion that if a conceptual schema is specifically meant, only *product data model* should be used. We could thus speak of a product data model for building descriptions. Ideally, the term *product model* should be reserved for the information base, the information about a particular individual artefact. We could thus speak of a product model of the White House, in much the same way as we would speak of a wire-frame or solids model of the same building, if some pure geometrical modelling approach was used.

In practice, derivatives of the term product model are often used in situations where the whole approach is implied, rather than a conceptual schema or an information base alone. If the whole approach is meant, it would seem easier to use product model alone, dropping the word data (e.g. *product model approach*, *product model based* rather than the more cumbersome product data model approach, product data model based).

Certain features are typical for all product data models. One of these is the large variety of object classes. Another is the fact that the objects can usually be arranged as pyramid-like decomposition hierarchies, starting at the top with one object describing the whole product and ending at the bottom with a large number of component objects. A third feature is obviously the need for information structures describing the shape and location of physical objects.

A building product data model is a particular kind of product data model, which structures information describing buildings. One prominent feature of building product data models is the explicit modelling of spaces. In this respect building product data models differ from traditional building descriptions using 2-D drawings and some geometrical modelling techniques. Building product data models share many information structures with product data models in general and are closely related to the product data models which are currently being developed for other industries such as off-shore and ship-building [Langbecker and Nowacki 1994].

The following two definitions sum up this section and provide the basis for understanding the rest of the report

Building product data model.

A type of conceptual schema where the universe of discourse consists of buildings throughout their design, construction, operation and maintenance. A building product data model models the spaces and physical components of a building directly and not indirectly by modelling the information content of traditional documents used for building descriptions.

Building product model

An information base describing some particular building. The structure of the information base is in conformance with a precisely defined conceptual schema (the corresponding building product data model).

4 REVIEW OF THE LITERATURE

4.1 PROBLEM DEFINITION AND EARLY PROTOTYPES (1970-1979)

This section presents a brief review of the research concerning building product data models. Since the research history in this field spans twenty years, a useful way of structuring the presentation is to divide the research into a number of distinct periods. Each of these periods is characterised by particular paradigms about the aims of the research. In addition, different software technologies seem to have dominated the interest of the researchers during the different periods.

The first rudimentary CAD-systems were developed in the early 1960's. Fairly quickly architects realised the potential use of such systems for architectural draughting and three-dimensional design [Mitchell 1977].

In the middle of the 1970's a handful of research groups in Great Britain and the United States started to develop computer-aided building design systems which tried to offer more than merely the automation of draughting work or geometrical modelling facilities. These development projects aimed at producing comprehensive integrated building design systems, which could be used throughout the design process from the briefing stage to detailed design. In addition to the generation of plan drawings and perspectives, the systems could assist in some evaluating tasks and could automatically produce a number of reports.

The most significant efforts from this period were the BDS, SSHH and GOAL systems in the UK and the GLIDE and CAEADS systems in the United States. Good overviews of the research during this period can be found in [Eastman 1978], [Bijl et al. 1979] and [Vanier and Grabinsky 1987].

Unfortunately, the computer hardware and software available at that time was not adequate for the task at hand, although some of the systems were even tested in real design projects. The prototype systems were both cumbersome to use and in many cases contained restrictions on possible designs and building components, which seriously limited their use in real construction projects. A good example is the BDS system which necessitated the use of a particular industrialised building system (the OXSYS method) and restricted designs to solutions composed of orthogonal building elements [Richens 1978]. The programming languages and database techniques that were used tended to lead to software that was difficult to modify and extend, and consequently inflexible to user demands for changes.

The research in this period did, however, result in some degree of consensus on the basic aims of such integrated building design systems. One of the earliest discussions can be found in [Eastman 1975].

4.2 COMPUTER-AIDED DRAUGHTING AND EXPERT SYSTEMS (1980-1984)

In the early 1980's, commercial draughting systems, which were usually marketed as computer-aided design (CAD) systems, came of age. The predominant hardware configuration of this period was the mini- or supermini computer running a number of dedicated CAD terminals. In the construction industry, CAD-systems were first taken into use by large multidisciplinary building design practices which could afford the price, which was more than the yearly salary of a designer per workstation. Typical methods for structuring data in these systems were overlay draughting and symbol libraries [Port 1984].

The growing commercial availability of these systems coincided with dwindling public support for development work related to integrated building design systems. During this same period, researchers started to take an interest in new software techniques emerging from artificial intelligence. Numerous projects were started to investigate the usefulness of expert systems for construction [Lansdown 1983], [Wager 1984]. In the early days before dedicated expert system shells and object-oriented environments became widely available, the Prolog language [Sterling and Shapiro 1986] was a popular software platform. The first generation of prototypes were typically small systems in both scope and the number of rules. Integrated building design was hardly the sort of problem for which this type of tool was suitable.

Researchers who studied integrated building design systems during this period, turned to relational database systems as a possible software environment [McIntosh 1984], [Autran and Florenzano 1985]. The resulting prototypes were still too slow and had too poor user interfaces to interest commercial software developers.

4.3 OBJECT-ORIENTATION AND CONCEPTUAL MODELLING (1985-1989)

4.3.1 The STEP standardisation effort

Towards the middle of the 1980's, advances in artificial intelligence (AI), database theory and programming languages [Brodie et al. 1985] started to provide powerful data representation methods which also seemed applicable to digital building descriptions. In a number of prototype projects, object-oriented methods were adopted. At the same time, the STEP standardisation process started and inspired many researchers to focus on the definition of conceptual schemas using formalised techniques, rather than on prototype development.

During the latter 1970's, the difficulty of exchanging data between CAD systems led to the definition of a neutral format for the exchange of geometry data, IGES [NBS 1986]. By the mid 1980's, most of the deficiencies and shortcomings of IGES, had been recognised. As a result, the development of a much more ambitious data exchange standard for product data, STEP, was started in 1985 [Owen and Bloor 1987]. STEP stands for Standard for the exchange of product model data and is organised under the International Organisation for Standardisation (ISO). The publications of the first draft standard version of STEP started in 1992. The standard currently contains 12 parts.

One vital part of the emerging standard is the information modelling language which was developed within STEP for formulating conceptual schemas: EXPRESS [ISO 1992c]. In background material for the standard, conceptual schemas can also be presented using the graphical modelling techniques IDEF1X and NIAM. Since the introduction in 1991 of a graphical subset of EXPRESS, EXPRESS-G, the use of IDEF1X and NIAM has decreased within the STEP community. An important part of the STEP standard is also the syntax for exchanging product model objects as sequential files [ISO 1992d]. This syntax is a general-purpose method for exchanging objects, whose information structures have been defined in EXPRESS.

Some national building product data model proposals, in particular the NICC [Tarandi 1993] and OOCAD [Serén et al. 1993] models, have included a data transfer syntax of their own. Other research projects outside STEP, for instance the European COMBINE project, have adopted the STEP object transfer syntax.

The overall framework architecture of the STEP standard has changed considerably since the publication of the first proposal in 1988. The most significant change has been the addition of the application protocol feature [Owen 1993] . This allows the use of selected, but precisely defined, subsets of the general purpose schemas of STEP (called resource models) for the conceptual schemas needed in narrow application domains (application protocols in STEP terminology).

The STEP organisation includes a working group for architecture, engineering and construction applications (AEC). In addition to buildings, this working group deals with conceptual schemas for ships, off-shore platforms, process plants and civil engineering works. Within this working group there is a separate subgroup particularly for building construction.

4.3.2 Formalised building product data model proposals

Two generic building product data model proposals, the GARM model and the Building Systems model, were discussed by STEP's AEC group during the years 1988-90. Despite the fact that neither one has been adopted as such as part of the standard, both models have had considerably influence on later efforts, as well within STEP as in research in this domain in general.

The General Architecture, Engineering and Construction Reference Model (GARM) was developed by Wim Gielingh at the Dutch research institute

TNO [Gielingh 1988]. The model was originally formalised using the IDEF1X notation, but later applications based on the model have predominantly been presented using NIAM and formalised in EXPRESS [de Waard 1992], [Gielingh and Suhm 1993], [Luiten 1994].

The GARM model organises construction process information at a high level of abstraction, regardless of the type of constructed artefact. It is applicable to buildings, as well as off-shore platforms, process plants, ships etc. The basic class is the *product definition unit* (PDU), which via subtyping, can be specialised to all other classes in the model.

PDUs can be specialised according to five different independent classifications. The classification according to the stage of the construction process, originally included seven stages (as required, as designed, as planned, as built, as used, as altered, as demolished). A subtype of the generic PDU was associated with each of these. The two subtypes *functional unit* (corresponding to the *as required* stage) and *technical solution* (corresponding to the *as designed* stage) have later been probably the best known features of GARM. Functional units are objects which collect the requirement attributes related to some part of a product, whereas technical solutions are objects describing chosen materials, connection details etc., which fulfil these requirements.

The AEC Building Systems Model was developed by James Turner at the University of Michigan [Turner 1990a]. The model was formalised using NIAM and illustrates a top-down development strategy based on systems thinking. In contrast to the GARM, the Building Systems model deals explicitly with buildings and tries to model the functional systems (enclosure, structural, mechanical etc.) that make up a building. The analysis is oriented towards the functions of building parts, not the technical solutions or materials chosen.

Two other ambitious projects to develop building product data model proposals, should also be mentioned at this point. The PROJIBAT project was initiated by the French building research institute CSTB in 1986 [Delcambre 1986]. A French conceptual modelling language called MERISE, which is close to the entity-relationship approach, was utilised. The final version of the model contained about 200 object classes organised in three hierarchical levels.

BIM (Bouw Informatie Model) was a co-operative project between several organisations in the Netherlands and aimed at simultaneously developing an activity model of the construction process and a building product data model. The methods used were IDEF0 for activity modelling and IDEF1X for the conceptual schema definition. The project produced two large manuals including several hundreds of diagrams [IOP-Bouw 1989].

The definition in 1986-1987 of the RATAS building product data model framework, coincided in time with the projects mentioned above [Ratas-committee 1988]. The model proposed the use of the entity-relationship model enhanced with inheritance for structuring a standardised conceptual schema for building descriptions. It divides the object classes that should be included in a fully detailed building product data model into five generic decomposition levels (building, system, subsystem, part and detail). Classification of relationships into two major categories (*part-of* and *connected-to*) was also proposed. The model was not formalised using a formal information modelling language, but was presented using example objects. Figure 4 shows an illustration of the structure of this framework model.

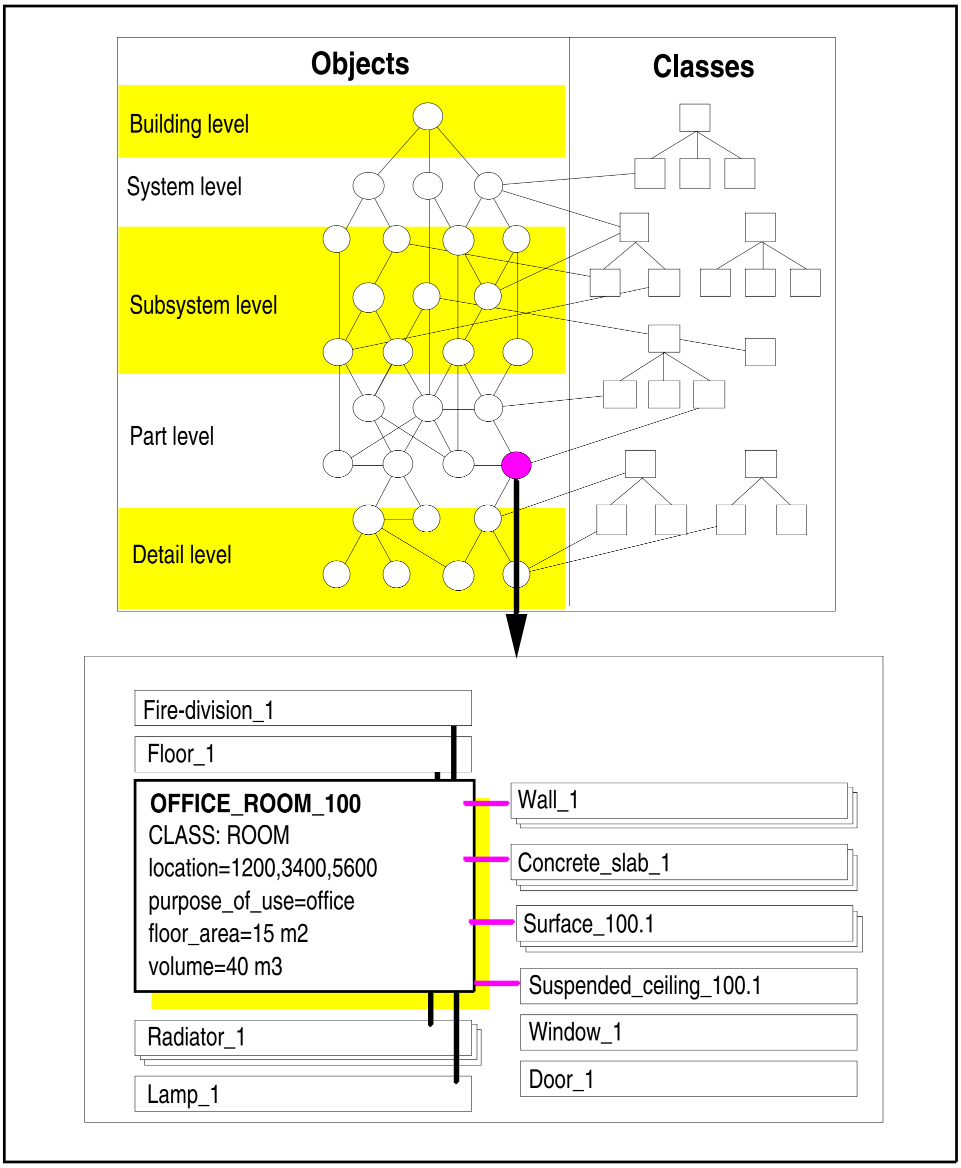

Figure 4. An illustration of the RATAS building product model framework. The upper part of the figure illustrates the decomposition (left) and generalisation-specialisation (right) hierarchies formed by the objects. The lower part of the figure "explodes" one object.

The KBS-model proposed in 1990 by the National Board of Public Building in Sweden, is a much more detailed model with a decomposition hierarchy quite similar to the RATAS proposal [Svensson 1991]. One feature of the

KBS-model is the relatively straightforward use of the existing Swedish standard for the classification of building elements as the basis for the definition of object classes.

4.3.3 Prototype-oriented research

In contrast to the more standardisation-oriented work described above, research groups at a number of French and American universities have built prototype environments testing methods for integrating different types of applications in building design. Projects of interest in France have included the X2A [Dufau et al. 1987] and TECTON [Hanrot and Quintrand 1988] systems. In the United States the *Data Model for Building Design* [Law and Jouaneh 1986] as well as the DESTINY [Sriram 1986], IBDE [Fenves et al. 1989], DICE [Sriram et al. 1989] and ICADS [Pohl et al. 1991] systems, should be mentioned.

In these prototype environments, emphasis was usually put on mechanisms for the integration of knowledge-based systems with CAD-systems, for instance using blackboard systems. Frame-based representations were predominantly used. The dominant paradigm at the American universities has been described in an article on the architecture of the KADBASE system [Rehak and Howard 1985] as “loosely-coupled databases”. The conceptual schemas needed for conversions between the internal data formats of the applications to be integrated, received less attention in the publications which reported on the research results. Nevertheless a building product data model at some level of detail had de facto to be defined in all of these projects.

4.4 LAYERED PRODUCT DATA MODELS (1990-)

4.4.1 A layered architecture of a building product data model

The work carried out during the previous period had highlighted the magnitude of the task of defining a comprehensive product data model which would cater to the needs of all parties involved in the design, construction and maintenance of a building. Most of the leading researchers now regard it as virtually impossible to define and validate such a model within one single research project. This task is currently regarded more as a research program for the scientific community in this domain in a perspective of several years.

The early pioneering research and activities within the STEP standardisation project had also highlighted the need for well-defined working methods in such definition work. The benefits of using formal information modelling languages had also clearly been demonstrated.

Recently, it has been proposed that it is more realistic to concentrate on the definition of less ambitious product data models which would cater to the information needs of applications in limited domains only. This implies a partitioning of the UoD into smaller sub-domains. Terms used to describe such models are *local product models* [Wright et al. 1992], *topical models* [van Nederveen and Tolman 1992] and *partial models* [Langbecker and Novacki 1994]. The European research project ATLAS has used the term *view type model*. The term *aspect model* is currently gaining acceptance and will be used below. The *application protocol* mechanism, as used in STEP, is one way of defining such aspect models.

If aspect models were defined independently for the different subdisciplines in building design and for the different stages of the construction process, we would achieve only slightly better integration than at present. An important part of the proposed strategies is thus also that these aspect models should be compatible with each other for those information structures which are common to several different aspect models [Luiten et al. 1991]. A term often used to describe such a schema is a *kernel* or *core* model. Such a strategy has recently been adopted as the basis for a working group within the STEP AEC committee which is planning and monitoring the development of a number of aspect models [Wix et al. 1994].

A version of such a strategy is used in the following as the basic framework for the presentation of this thesis. The set of information structures which makes up a fictive comprehensive building product data model is partitioned, using an onion-like overall architecture. The number of layers in such an architecture is obviously to some extent a matter of choice. In the opinion of this author, decomposition into five layers would seem appropriate. These five layers are shown in table 2, which also tries to place some of the most significant models discussed in the literature, in the different layers.

The application models of the outermost layer consists of the conceptual schemas which are implicitly or explicitly used in individual applications. The next layer consists of a considerable number of aspect models which concentrate on the information needs of particular phases and/or subdisciplines in a construction project. The innermost layers contain generic information structures, which via inheritance are shared by models residing in the outer shells. Three such layers are included in the framework used in this article: the information modelling language, the generic product data model and the building kernel model.

The advantage of such an architecture is that it becomes possible to define product data model structures incrementally, step-by-step, and to integrate relatively independent models from the different layers with each other. In the following, the different layers are discussed one at a time.

Table 2. A layered framework architecture for grouping the information structures which are used in building product data models.

Model layer	Information structures	Proposed models
Information modelling language	Objects,attributes, relationships, generalisation-specialisation, rules, methods, messages	Relational data model Entity-relationship model The object The frame
Generic product description model	Aggregation-decomposition, type objects, versions, shape and location information	STEP resource models EDM GARM OOCAD
Building kernel model	Generic information structures which are common to most disciplines and phases in design and construction of buildings	Building systems model GSD-model RATAS-framework KBS-model, NICC model
Aspect models	Information structures particular to a specific design or construction discipline and/or phase in the construction process	COMBINE IDM RATAS prototypes Structural Steel Model CIMSTEEL LPM
Application models	The conceptual schema of one particular application	(Not subject for research)

4.4.2 Information modelling language

As already discussed in section 3.2, there is a wide range of alternative information modelling languages to choose from. At least the following generic approaches and particular information modelling languages have, to this author's knowledge, been used or proposed for formalising building product data models.

- The relational model
- The structural model (an extension of the relational model)
- The Entity-Relationship approach
- The Entity-Relationship approach with inheritance

- NIAM
- IDEF1X
- EXPRESS
- The OMT method
- Semantic nets
- Predicate logic
- Frames
- Objects according to the OO programming paradigm

Of the information modelling languages or approaches proposed in the research literature, the Mole system (Modelling Objects with logic Expressions) developed at EdCAAD is close to a frame [Bijl and Szalapaj 1984]. The frame has also been a very popular data model in prototype work at American universities. The RATAS framework model is based on the entity-relationship model enhanced with generalisation-specialisation [I]. Law and Barsalou have proposed using the structural model, an extension of the relational data model, as the fundamental modelling approach [Law and Barsalou 1989] .

There is currently a discussion as to which information modelling capabilities are required for the definition of building product data models. Features which are being discussed include multiple inheritance, dynamic reclassification of objects, inclusion of messages etc. [Froese et al. 1993]. Some of these features are more related to the internal schema aspects of how to store and access data than to modelling the Universe of Discourse on the conceptual schema level.

4.4.3 Generic product data model

The second layer contains information structures which are particular for the description of artefacts designed and manufactured by man, but which usually would not be needed in database applications for bank accounts, personnel information systems etc. Due to its industry-independent nature, the resource parts of the STEP standard deal predominantly with information structures in this layer.

The information structures in this layer typically model the shape and location of objects, assembly-part relationships between objects, versioning of design solutions and the reuse of standard components used repeatedly in the same product.

Of the models presented earlier, GARM is primarily situated on this level. The OOCAD generic product data model proposal, is also a good example of a model on this level [Serén et al. 1993].

One recent model which is situated somewhere between the information modelling language level and the generic product data model layer, is the Engineering Data Model, EDM [Eastman 1992]. This model uses sets and first-order logic as its modelling approach and emphasises the modelling of constraints between objects. The primary object class in EDM is the functional entity. One particular feature of EDM is that it models composite objects or assemblies using a set of constraints called accumulations. This enables quite detailed modelling of the knowledge included in different construction technologies. EDM is proposed as an alternative to the data-centred integration methodology of STEP. In addition to static data, EDM would be able to integrate the modules of knowledge present in different design and construction applications [Eastman et al. 1993].

4.4.4 Building kernel model

The building kernel model contains information structures explicitly needed for the modelling of buildings, as opposed to other products such as machines or cars. This layer should only contain information structures which are common to a large number of applications from the different subdisciplines and phases of the overall construction process. A good example of a model predominantly situated in the kernel layer is the Building Systems Model discussed above. This model analyses a building top-down and dissects it into its functional systems and their parts. The RATAS framework model is also an example of a highly generic model partly situated in this layer.

A different strategy for modelling information pertaining to the kernel has been used in a number of French research projects and in particular, in a synthesis made from the conceptual models developed in them [GSD 1991]. According to this strategy, the analysis starts bottom-up by defining typical building components (spaces, walls, windows, doors etc.) and the network-like relationships between them.

One approach advocated by a number of researchers could be described as the minimal approach [De Vries 1991], [Tarandi 1993], [Serén et al. 93]. In this approach the size of the kernel is kept extremely small in order to make it more flexible and adaptable to the needs of different participants in the construction process. Different types of building components (slabs, columns, windows etc.) are not modelled as separate classes in the schema, but are included as the values of a *building component type* attribute of a generic building component class. This kind of approach is currently receiving increasing attention.

4.4.5 Aspect models

The aspect model layer contains any information structures which are aimed at the needs of particular subdisciplines and or phases in design and construction. Aspect models typically contain many information structures from the models of the first three layers, using inheritance. The layered architecture discussed here is only a sort of view mechanism for sorting information structures into different categories depending on where they have originally been defined and where they are used. If the class *wall* is part of the building kernel model, the fact that walls have a *u-value* could be part of the aspect model for HVAC design. It would obviously not make sense to process information about u-values without specifying to which walls they are related. Similarly a *column* class would be part of the kernel, whereas the details of its reinforcement should be modelled in some aspect model for structural engineering.

The Integrated Data Model (IDM) of the European COMBINE project, provides a good example of this category of conceptual schema [Dubois et al. 1992]. The model is documented using NIAM and EXPRESS and contains approximately 400 classes integrating the data needs of six applications related to the energy-conscious architectural and HVAC design of buildings. Other good examples of aspect models are the structural steel framing data models developed at Stanford University [Lavakare and Howard 1989] or by the European CIMSTEEL project [Watson 1995].

4.4.6 Application models

The last layer in the framework includes the implicit or explicit conceptual schemas which have been implemented in individual computer applications. This layer is included in the framework for illustrative purposes only, not because of a need to develop standards or well documented reusable schemas, as is the case for the other layers.

The relationship between application schemas and the schemas in the other layers can either be direct or indirect. In a direct relationship, the application schemas are created from standardised kernels and aspect models or from library schemas using inheritance by taking appropriate classes and adding the necessary “private” information structures. This could offer considerable benefits in terms of quality assurance of software and programming efficiency.

In an indirect relationship, conversion software is needed to create neutral transfer files or to store information in some centralised database separate from the databases of the individual applications. This approach would typically be used if already existing software is being integrated.

5 MAIN RESULTS

5.1 MAIN RESEARCH ISSUES

The five separate articles which are included in this thesis contain detailed discussions of the results of the study. In particular, one of the articles gives a fairly comprehensive presentation of the four prototypes that were developed [III]. The presentation is not repeated in this summarising report. A table with short descriptions of the prototypes and their information bases is shown below (Table 3).

Table 3. The four prototypes developed during the study used different combinations of software tools and different information bases.

Proto-type	Data storage media	User interface	Case material	Classes/ Objects
1	Relational database	SQL-queries	One floor of an office building	20 / 2000
2	Hypermedia	Hypermedia	Two rooms	10 / 40
3a	Relational database	Hypermedia	Two rooms	30 / 75
3b	Relational database	Hypermedia	Health centre	11 / 1000
4	CAD-system & relational database	CAD-system, hypermedia	Office building	50 / 1000

This presentation of the results of the study is divided into three sections. The first concerns the requirements which building product data models can be expected to fulfil. The second discusses a number of important information structures needed in building product data models. The third concerns the integration of building product data models with conceptual schemas for other types of construction process information. The presentation is partly structured according to the kernel-aspect model framework, which was discussed earlier in section 4.5. It should be stressed that only a selection of the results are included in this presentation.

5.2 REQUIREMENTS FOR BUILDING PRODUCT DATA MODELS

5.2.1 Five main requirements

One important task in the study was to examine the requirements that can be posed on building product data models. A discussion of such requirements can be found in the first article included in this thesis [I, p. 72].

Requirements posed on building product data models can be formulated from two different viewpoints. Firstly, all general requirements for conceptual schemas, as outlined in section 3, apply by definition to building product data models as well. Secondly, requirements can be formulated based on an analysis of the shortcomings of current construction documentation practice. The requirements discussed below are of both types.

It should be noted that the first formulation of these requirements [I, p. 72] was done for a single comprehensive building product data model (one conceptual schema). The architecture of a building kernel model supplemented by aspect models, had not yet been proposed at that time. The requirements apply almost equally well, however, to the layered framework discussed in section 4.4. Some modifications are nevertheless needed to take into account the fact that the scope of a kernel is smaller than the scope of the comprehensive building product data model originally intended and the possibility of numerous partly overlapping aspect models.

In the following, the requirements, as well as examples from the prototype work which demonstrate their fulfilment, are presented and discussed. The phrasing of some of the requirements has been slightly changed, compared to the original formulation as presented in [I, p. 72]. The author feels that the formulations below are clearer and better reflect the originally intended meaning of the requirements. In addition, the formulations have been changed in order to apply to the kernel-aspect model architecture.

A building kernel model should

- be capable of modelling all possible types of information used for building descriptions
- cover the information needs common to applications in all phases and subdisciplines of the design and construction process
- be non-redundant in its definition of information items
- be capable of supporting alternative presentation formats

- be independent of restrictions on permissible information structures posed by the limitations of application software.

An aspect model should

- be capable of modelling all possible types of information used for building descriptions within a particular phase and/or subdiscipline
- cover the information needs common to a limited number of applications in some phase and/or subdiscipline of the design and construction process
- be non-redundant in its definition of information items internally and in relation to the kernel model
- be capable of supporting alternative presentation formats
- be independent of restrictions on permissible information structures posed by application software.

Each one of these requirements is discussed in the following.

5.2.2 Capability for modelling any information type

This requirement states that the kernel and the aspect models should be *capable of modelling any type of information* describing a building as a product. This formulation is in fact less ambitious than the original formulation, which stated that a building product data model should *structure all the information* needed to construct and maintain a specific building.

The original requirement stated that the UoD for the conceptual schema should be all the information needed about a building during design, construction and maintenance. The rewriting implies that the information modelling methods should be such that the schema can easily be extended to cover any parts of this whole universe of discourse. This requirement has been included specifically because traditional building documentation splits design information into a number badly interconnected categories, for example drawings containing the geometric description of the building, specifications describing materials and working methods, the brief containing the clients requirement for the extent and quality of spaces. One primary aim of product models is to describe all this information in a uniform way. It should consequently be possible to use product model data for the automatic production of any such documents, or as input information to analysis applications which traditionally rely on the human interpretation of, or computerised interfaces to, such documents.

The prototypes developed during the study illustrated this principle by modelling different categories of information and by showing how different types of documents and user interfaces can be produced from the product models. Traditional types of information that were modelled included the shape, location and material properties of building components. An information category which in traditional documents is only implicit, but is included in building product models, is the relationships between objects.

The second prototype demonstrated in particular, the ability to mix the data needed for producing drawings and building specifications in the product model description [II, pp. 183-184]. It also allowed for viewing the data in a format called a *conceptual window*, which especially utilises relationship information [III, p. 273]. This is a presentation format which is easy to implement in hypermedia software, but which cannot be found in traditional documents.

5.2.3 Coverage of the information needs of different phases and disciplines

This requirement deals with the extent of the universe of discourse and states that the kernel should structure information which is applicable throughout the whole design, construction and maintenance process as well as across engineering subdisciplines (figure 5 illustrates this principle). There should thus not be separate information structures in the kernel for the same information, even if the information today is contained redundantly and in slightly different formats, in separate documents used in different tasks of the construction project.

One typical example of the problems in today's practice, which this principle tries to remedy, is the recording of space-related information. Spatial needs are often in the client's brief recorded using a word processor or spreadsheet, which is difficult to relate to the implicit rooms indicated by enclosing walls in a CAD-drawing. In the product model approach, the object class for describing spaces is quite central. This class is already highly relevant during the briefing stage, where the client's needs are expressed as requirements for different types of spaces, and at the other extreme, during building maintenance, where the grouping of information about surfaces and their conditions is often space-centred. The space is a central class in all the prototypes of this study, as well as in the conceptual schema for spaces, space boundaries and enclosing structures [IV].

Aspect models, on the other hand, are related to the information needs of limited numbers of applications used either in specific phases of the project or specific subdisciplines, or some combination of these. Typical examples of object classes which can be found in aspect models are fire zones, u-values and structural reinforcement details. These classes relate to other items which are modelled in the kernel, such as spaces and walls, but need not be in the kernel themselves.

The prototypes did not originally aim to demonstrate the use of aspect models. In retrospect, the version of the third prototype used for the energy-conscious design of a building [Talonpoika et al. 1990], was an example of an aspect model, since the model only contained the information that was relevant as input data for energy calculation software. The fourth prototype also implicitly contains an aspect model, for the management of quantities in the tendering stage.

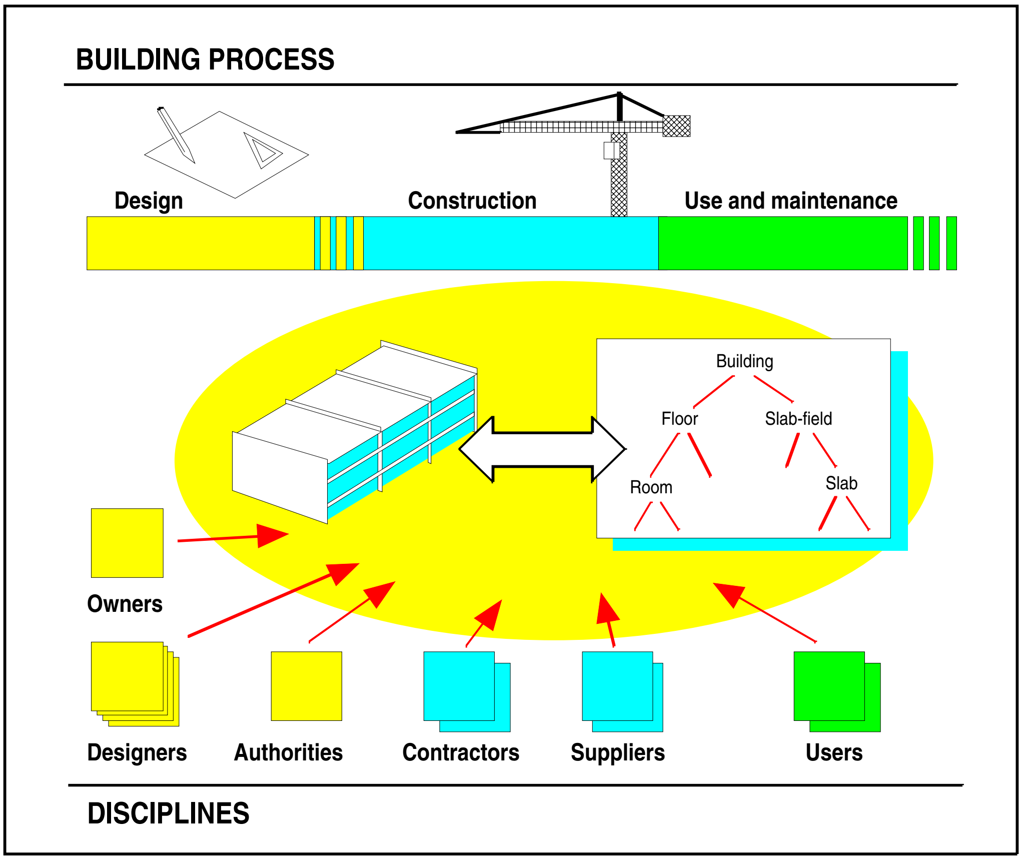

Figure 5. The building product model kernel should specify the common information structures which are applicable throughout all the stages of the building process and across all the subdisciplines involved.

5.2.4 Non-redundancy of information structures

It seems natural to include a requirement that the same information item should be defined only once in a conceptual schema for a UoD. This is in line with the fundamental principles of conceptual modelling. However, the architecture of a kernel model supplemented by a large number of aspect models, does have implications for how strictly this principle can be applied to the total system formed by the kernel and the aspect models. Within the kernel model, the principle can be applied as such. The same applies to any aspect model viewed in isolation, or to the union of the kernel model and any single aspect model in isolation. The principle does not, however, apply to the union of several aspect models, which may contain redundancy. The primary reason for this is that it would not be realistic to assume that the definition of all aspect models which may be needed for the construction industry, could be so highly co-ordinated that their schemas could be fully harmonised. (If such harmonisation could be achieved, the dream of a comprehensive building product data model would de facto be fulfilled.)

Questions about how to handle functional interdependencies between different information items were raised early on in connection with the original formulation of the principle of non-redundancy [I, p. 72].

Complications arise in situations involving different, but highly interrelated, information items which can be linked together using derivation rules. It would be possible to define conceptual schemas which include these derivation rules, as parts of the class definitions. In principle, such rules could then be hardwired into database applications as integrity constraints in the same right as more straightforward integrity constraints (such as a cardinality rule that each door serves exactly two spaces).

Rules such as the one calculating the area of a space from the locations of the walls that surround it, could be allowed in aspect models (for instance for quantity take off applications). The inclusion of such rules in a kernel could, however, cause complications and should be avoided. The reason for this is that different aspect models may use different subsets of the kernel, via inheritance. An aspect model for briefing or for facilities management could, for instance, reuse the space class of the kernel model, but might leave out the classes for walls altogether. In such a case, the derivation rule attached to the area attribute would be indeterminate, since the necessary information to evaluate the rule would be missing.

5.2.5 Capability for supporting alternative presentation formats

The structure of the kernel should not emulate the implicit conceptual schema of any currently existing document type (which is what draughting and word processing applications do). In this sense, the structure of both the kernel and the aspect models should be independent of the output documents produced from product model data. On the other hand, the information content of the conceptual schemas should naturally provide an important part of the information required to produce most of the documents needed in the design and construction process.

This requirement is in fact just a rewriting of the ANSI SPARC three-layer framework, which distinguishes between the one conceptual schema and several possible external schemas. The latter correspond to user views on the information contained in the conceptual model.

Due to the need to convince practitioners of the potential benefits of the product model approach, a great deal of emphasis in the prototype work was put on illustrating the possibilities for producing different types of documents or user interfaces from the product model data (Figure 6). This aspect of the prototypes did in fact prove to be a key factor in convincing people from the industry of the validity of the whole approach.

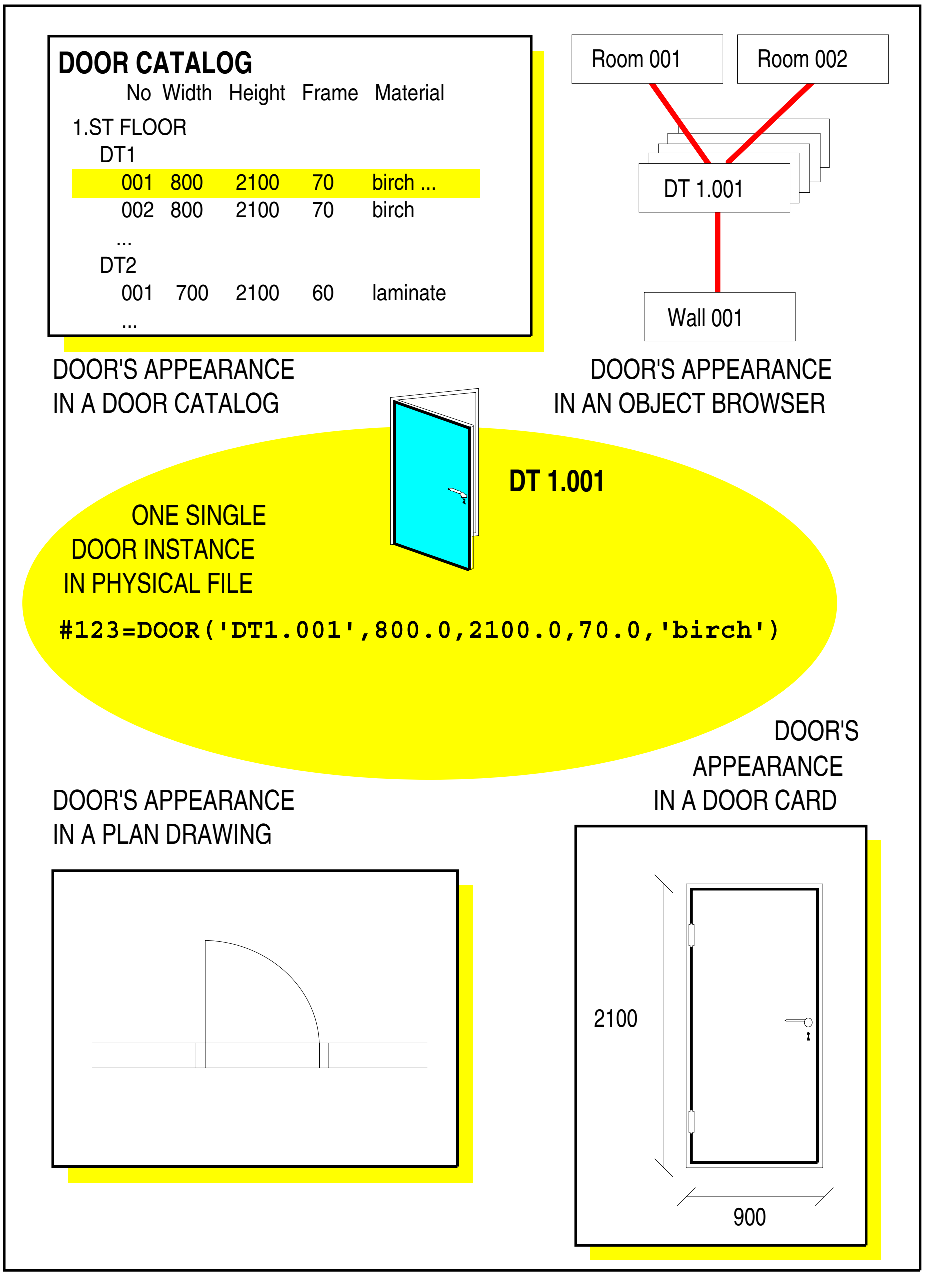

Figure 6. The prototypes which were developed during the study demonstrated how different kinds of documents and user views can easily be produced from a product model representation of the building.

The four prototypes include outputs which highly resembles current construction documents, but also innovative user interfaces which take advantage of the information content of the product model representation. The former include building specifications and bills of quantities [II, pp. 180-181], [III, pp. 275-276]. The latter include hypermedia interfaces to building objects and the abstraction hierarchies these form, as well as object class and type object browsers [II, pp. 183-184], [III, pp. 274-277]. The room-card which was produced in the project is an interesting new type of document, which is very straightforward to produce from product model information, but would be tedious to draw manually for each particular room.

5.2.6 Independence from restrictions of application software

This requirement is a consequence of the separation of the conceptual schema from the internal schema in the ANSI SPARC framework. The conceptual level should be concerned with modelling the logic of the UoD only, whereas the internal level relies on the data structuring capabilities of particular database systems or programming languages.

The independence from the fundamental data structures of software used in applications is, on closer inspection, not strictly categorical. Clearly, it would be more straightforward to build prototypes with programming tools based on paradigms closely related to the methods used in conceptual modelling, than with programming languages such as FORTRAN or commercial CAD-systems. Object-oriented programming and object-oriented databases are consequently increasingly used in prototype work.

It is, however, with some limitations, also possible to build prototypes using more conventional tools. In particular, if CAD-systems are used, some parts of the prototypes are obtained with little programming effort, and the focus is on implementing interesting aspects of the conceptual schemas.

The strategy in this study was explicitly to use off-the-shelf commercial software that runs on microcomputers, such as CAD-systems, relational databases and hypermedia software, for the development of the prototypes. Consequently, the results of the prototype work illustrate the possibilities and difficulties in implementing the information structures of building product data models using these categories of software only.

On the whole, the prototype work illustrated that it is possible to develop building product model applications using quite conventional software. The problem is, however, that certain structures which are very natural to use in the conceptual models are difficult to implement using these categories of software, and that this may lead to detrimental compromises in the definition of the conceptual models themselves. Examples of information structures which caused problems for the kinds of software used in this study, were inheritance, many-to-many relationships and type objects with flexible allocation of data values.

The conclusion from this study is that no concessions to implementation issues should be made in the conceptual modelling work. The information structures in the conceptual schemas should be based only on the logic of the universe of discourse and not limited by the particular data structuring mechanisms of the prototype tools used.

5.3 INFORMATION STRUCTURES IN BUILDING PRODUCT DATA MODELS

5.3.1 Choice of information modelling language

The primary reason for the large number of different conceptual modelling approaches used for defining building product data models, is the variety of aims for such work. Some researchers want to develop standards for data exchange only, while others want to focus on developing tools which support designers in the more creative aspects of their work.

The following viewpoints seem to have had some significance on the choice of information modelling language (or the subset of features actually used in some particular schema):

- Capability for modelling the semantics of the universe of discourse without simplifications caused by the information modelling language.
- Capability for modelling the designer's intents and aims.
- Support for the evolutionary process of design (extendibility of the schema).
- Usefulness for the exchange of data between heterogeneous computer applications in construction.
- Technical feasibility for implementation using current commercial software.
- Realistic possibilities for achieving standardisation (in terms of reaching consensus in standardisation bodies and expenditure).

Emphasis of the first three viewpoints would seem to lead to the use of more complex information modelling languages, whereas emphasis on the last three would seem to indicate the use of less complicated languages.

The RATAS framework model proposed the use of the entity relationship model enhanced with inheritance between classes. Such a model seemed to contain sufficient semantic power for the task at hand. At the same time, a relatively simple modelling approach was recommended in order to achieve standardisation.

A step-by-step strategy for the gradual extension of the information modelling features used in construction building product data models was proposed in the second paper included in this thesis [II, p. 177-178]. This proposal was mainly based on considerations of what would be possible to standardise and seems still to be valid. Since most modelling approaches are relatively overlapping in the information modelling mechanisms they contain (as discussed in section 3.2), it would make sense to start any standardisation effort with the features which are common to most approaches. Thus the first step would be to choose the main object classes for representing building parts and the systems they form. The second step would be the definition of the most important attributes of these classes. The third step would be adding major relationships needed by many applications. A further layer of extensions could include modelling more complicated integrity constraints.

5.3.2 The abstraction hierarchy

One of the principal features of the RATAS framework model was that it proposed a generic abstraction hierarchy for a building product data model. The hierarchy consisted of objects on five specific levels: the building object, building system objects, subsystem objects, part objects and detail objects. These would typically be generic classes included in the building kernel model.

The RATAS model was primarily a *decomposition hierarchy*, using so-called part-of relations between objects. Thus objects from lower levels in the hierarchy would be parts-of higher level objects. Another way of looking at the abstraction hierarchy in a product data model, is as a *specialisation hierarchy* using subtyping of object classes. In this view, the root of the classification is the building product model object class, which as subtypes, would have generic objects for the five main abstraction levels. These could then be further specialised in more detailed subclasses.

One deficiency in the original framework model which became evident especially in energy-related projects such as COMBINE, is the restriction of the objects only to ones describing the building itself and its internal decomposition into parts. Many applications also need information about the immediate surroundings of a building, and how a building and its parts relate to these surroundings. Consequently, it is proposed that objects for the physical building site and for the concept of the total system formed by a site and the buildings on it, should be included in the kernel [Björk et al. 1991b, pp. 61-62]. The site object would contain information about things which exist on the site regardless of buildings, for instance climate. A site would also form an abstraction hierarchy of its own, including natural objects such as soil layers and man-made objects such as parking spaces and sewage piping.

In addition, the final report of VTT’s product model project proposes the inclusion of separate generic classes for subspaces and joints [Björk et al. 1991b, pp. 65-67]. The inclusion of subspaces is in line with the model for spaces, space boundaries and enclosing structures [IV]. The inclusion of a generic class for joints is motivated by the fact that joints of different kinds are extremely important in construction.

A proposal for the extension of the abstraction hierarchy of the RATAS framework model is shown in the following EXPRESS-G diagram (figure

7). Note that subsystem objects can be part of other subsystem objects (modelled by recursive decomposition).

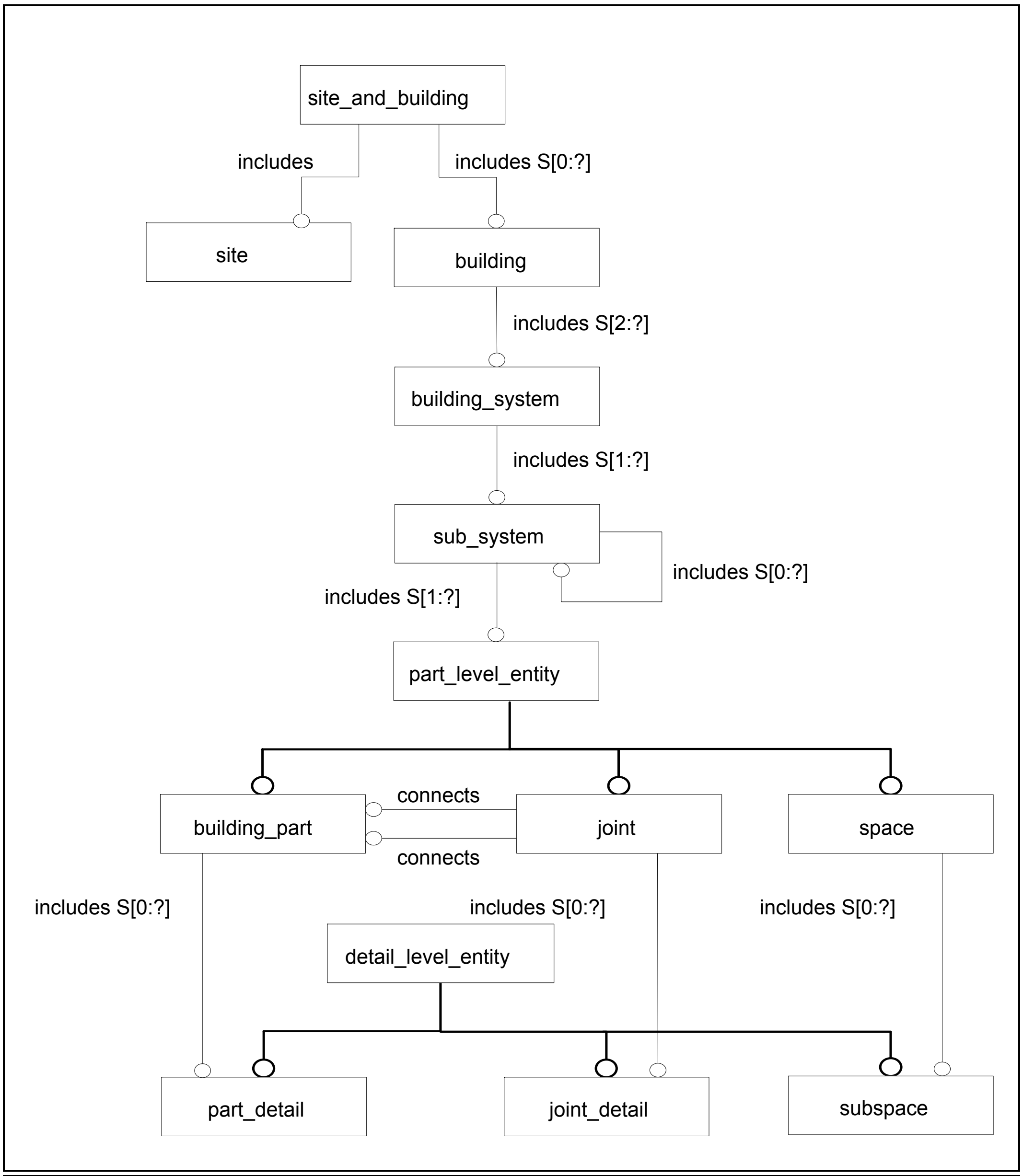

Figure 7. The proposed extended abstraction hierarchy of the RATAS model.

Towards the end of the study, modelling low-level attributes received increasing attention. Low-level attributes can typically be used for expressing the properties of tangible building components (i.e. length, volume, colour, price). In many conceptual schemas such attributes are directly defined as primitive or data types (text, integers or real numbers).

The need for modelling properties of building components as composite data types having definitions of their own in a schema, was demonstrated by experiences with the fourth prototype. A project (in which the author was involved) which tried to define a conceptual framework for information structures to be used in quality systems for the construction industry, also demonstrated the potential benefits of such an approach.

In the EXPRESS language, this would be handled using ENTITY or TYPE declarations for such properties. Any time such a property would be modelled as the attribute of a class representing a building element the data type of this attribute would be the class or type in question. A major benefit of this strategy would be in harmonising different applications which would rely on such predefined attribute classes. Good examples would be properties used for modelling cost, time, weight etc. Some general guidelines were consequently included in the final report of the product model project [Björk et al. 1991b, pp. 74-77].

One of the advantages that this strategy would offer is the reusability of data structures from standards such as the resource parts of STEP. Another is the possibility to utilise more traditional classification systems dealing with this type of building information. An example, for instance, is the CIB master lists for structuring documents relating to buildings, components, materials and services [CIB 1972].

So far, this strategy for modelling low-level attributes has not been tested by prototype work, owing to the limitation of the tools used. It is clear that fully object-oriented programming languages and databases are needed for efficiently implementing this method of modelling.

5.3.3 The type object mechanism

The notion of a type object is an information structure of interest to any developer of product data models, regardless of domain. In our multilayered framework, type objects would be situated in the generic product data model layer. Experiments with the type object mechanism were carried out specifically in the fourth prototype [III, pp. 276]. In addition, the final report from VTT's product model project, contains a longer discussion [Björk et al. 1991b, pp. 80-84].

The most general definition of a *type object* is a product model description of anything ranging from a whole building to a brick, where some information in the description is in an undefined state, such as the location of instances in a particular design. It is important to make a clear distinction between the concept of an object class on the one hand, and the concept of a type object on the other. A type object is something which exists in an information base, not in the conceptual schema. The type object is itself an instance of an object class defined in the conceptual schema. The occurrences of a type object are similarly instances which exist in information bases (we use the word *occurrence*, in line with other researchers [Gielingh 1988], [Serén et al. 1993]). The type object is used by inserting a pointer of some sort in each occurrence object.

The choice of the term type object is somewhat controversial, since a “type” has an accepted meaning in programming language theory [cf. Watt 1990]. Type object is derived from the construction industry use of *type solutions* (predefined technical solutions for wall structures etc.) in current documentation practice. Since the use of this term is accepted among building product model researchers in Scandinavia and has been used extensively in the earlier articles [I-V], it has not been changed. Alternative terms could be *template object* [Boman et al. 1991], *specific object* [Gielingh 1988].

Existing CAD-systems already contain mechanisms which, to some extent, function in the same way as type objects. The ability to store predefined symbol libraries, which the designer can create occurrences from and position in his drawings, helps to save storage space and facilitates the designer’s work. In product model applications, type objects play a similar role. Each time the designer wants to use some solution which has been defined previously, he is able to create an occurrence of it and will only need to input the information which specifically differs from the original (an example is shown in figure 8).

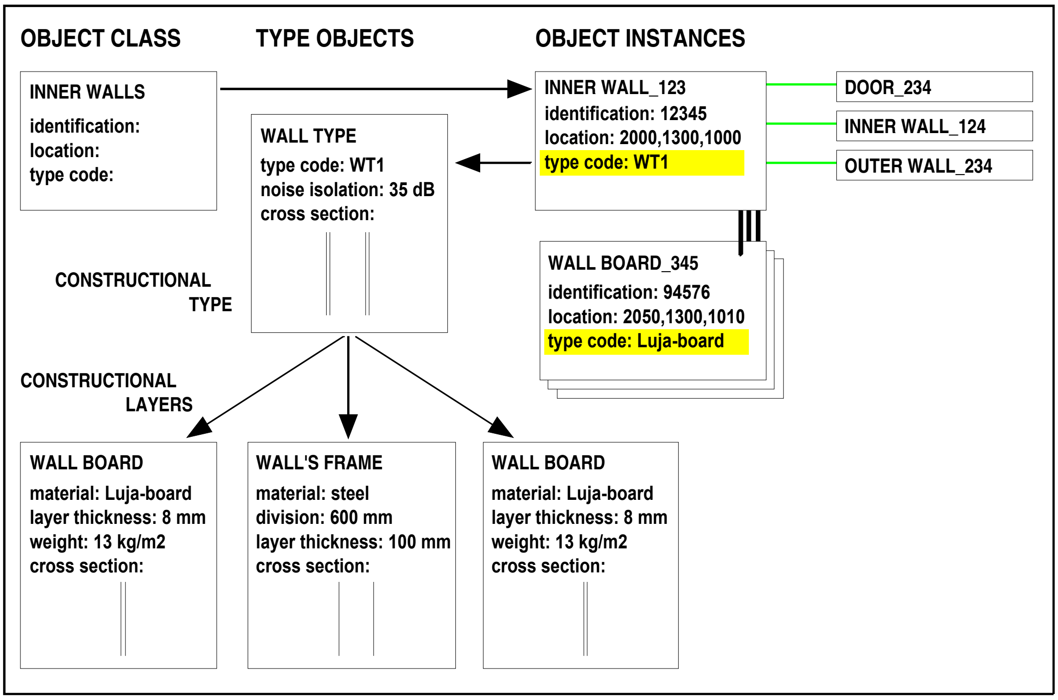

Figure 8. Type solutions are already used in today’s practice. Product models should make it even easier to retrieve, use and adapt earlier solutions to design problems.

There is usually no need to define a large number of separate object classes for type objects in a product data model. This is due to the fact that the type object and its instances both follow the same conceptual schema. Thus one option would be to consider the definition and implementation of type objects an issue for the development of applications, rather than an information structure to be explicitly included in building product data models.In the opinion of this author, however, the concept pair *type object-occurrence object* should be defined on the generic product data model level in the four-layer architecture we have adopted. The main reason is the need to transfer type objects between applications, sometimes even separately from any occurrences. A very likely trend is for developers of general construction databases to start to use product model methods for structuring data about materials, prefabricated parts, construction methods etc. In effect, such databases would then become libraries of type objects.

Experiences of implementing simple type objects were gathered with the fourth prototype in this study [III, p. 276]. This prototype exemplified how the limitations of the available prototyping tools imposed a solution on the conceptual schema which was found to be too restrictive. In the prototype, all objects in the actual product model description of a building were, by definition, occurrences of some type object within the same model. All other data except the exact position of the object and its identifier, were defined on the type object level only. This data structure was relatively easy to implement using the symbol handling facility of the CAD-system and the tables of the relational database. Program routines for taking off quantities were programmed in a way which highly resembles current methods in the construction industry.

A rigid type object structure of this kind was found to be relatively well suited for handling information about doors, windows and other categories of objects which usually occur with fixed dimensions and which can be modelled as simple type objects. The overall usefulness of the type object mechanism was clearly illustrated by this prototype. On the other hand, modelling certain object classes of varying dimensionality, such as walls, presented problems. For each wall occurrence of a different length, a new type object had to be created, since the wall's dimensions could only be given at the type object level. (One way to solve this problem would be to separate the wall's cross-section description into an object of its own, separate from the main wall object.)

A conclusion is that we need more flexible mechanisms for dealing with type objects, especially complex type objects with internal decomposition hierarchies. The mechanisms should preferably enable the designer (not the conceptual modeller or application developer) to decide independently for each type object-occurrence object pair, what data to store in the type object and what data in the occurrence object.

5.3.4 Modelling spaces and enclosing structures

VTT's product model project, as well as the author's involvement in the COMBINE project, clearly identified a need to include the topology of spaces and the structures that surround them in a future building kernel model. There are numerous applications in construction which need information about spaces and how they relate to walls, wall surfaces, openings etc. The four prototypes developed in the study relied heavily on this kind of data for facilitating object browsing and queries to the product models.

In addition to the prototype work, this part of the study included a theoretical analysis, comparing the implicit conceptual schemas of the study's prototypes with the schema that was defined in the COMBINE project [Dubois et al. 1992], as well as with two other schemas that had recently been published [GSD 1991], [de Waard 1992]. An important aspect of the work was the use of both the EXPRESS language and its graphical subset EXPRESS-G. A data dictionary containing natural language definitions of all the object classes was also produced. This was motivated by the author's personal experience of the fact that endless discussions seem to emerge in conceptual modelling work about the meaning and exact definitions of concepts.

The resulting conceptual schema contains 28 object classes [IV]. The complete model is shown in figure 9. The key feature of the model is that it concentrates on modelling the functional relationships between the elements that form spaces and their enclosures. Examples of relationships are thus *bounded by*, *has surfaces, forms* etc. The relationships are purely functional and no geometrical modelling concepts are used as intermediaries for implicitly carrying this information. Consequently, the model can be combined with any methodology for modelling the shape of different building elements.

Another important feature is the inclusion of highly generic classes for assemblies of spaces, space boundaries or enclosing entities. In aspect models, such assemblies could then be subtyped into apartments, fire zones, building facades etc. The model also contains classes such as subspaces, which are spaces formed by functions or user activities within fully enclosed spaces.

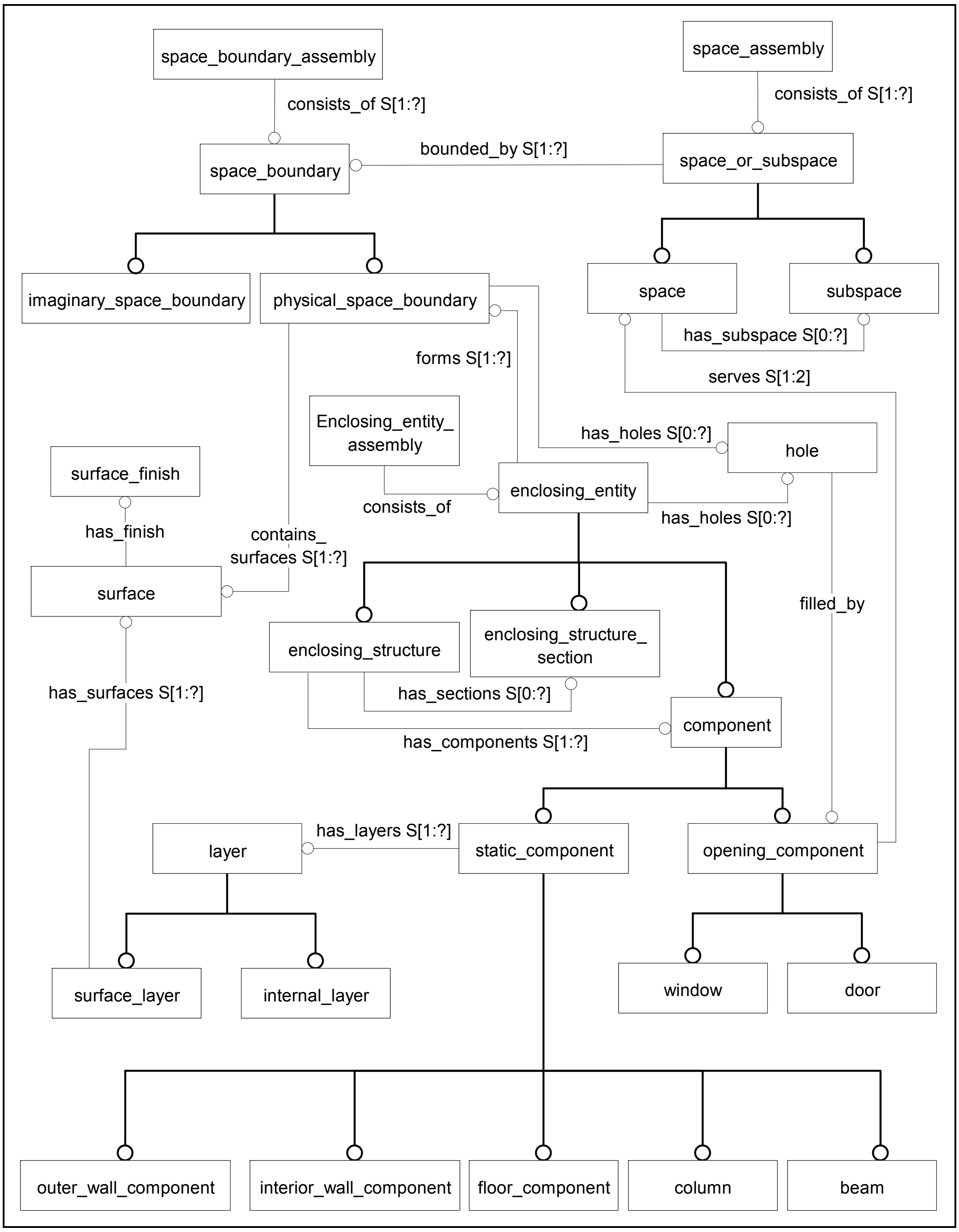

Figure 9. A conceptual model of spaces, space boundaries and enclosing entities.

5.4 INTEGRATION WITH OTHER CONSTRUCTION PROCESS INFORMATION

One of the aims of the study was to investigate the relationship between product model information structures and other information of relevance in construction. In particular, the relationship to traditional construction classification systems received attention, since the existing Finnish standard was under revision in 1988-90 [Talo 80 group, 1988], simultaneously with the study.

Apart from building description information, other important information categories used in the construction process include information about cost, construction process activities and the logistics of building materials. In all of these areas there is a growing interest in modelling of information structures [Björk 1994a]. Unfortunately, the techniques that are used (conceptual modelling for product data models, graphical activity modelling methods such as SADT [Marca and McGowan 1987] for construction process models, EDIFACT message definition for electronic trade and traditional hierarchical coding systems for information classification systems) are incompatible. There is thus an urgent need for a generic conceptual schema of construction process information, which could act as a framework for the development and integration of more specialised schemas. A generic schema of this type would facilitate the integration of product model based CAD-applications with other software used in construction.

The article *A Unified Approach For the Modelling of Construction Process Information* makes an attempt to show the elements of such a framework schema [V]. The IDEF1X method was used in the article. A later version of the model rewritten in EXPRESS-G is shown in figure 10.

An important proposal put forward in the article [V, p.184] is that any object that is created in an application during a construction project should be tagged with an identifier which is unique over the whole project and remains the same even though the object description varies (versioning). This applies equally to objects describing building parts, particular activities in scheduling software, or objects describing materials that are used as resources. The principles used for defining the identifiers are irrelevant provided that they guarantee uniqueness. It is of crucial importance that the same identifier remains with the object as it is transferred to different applications downstream in the design and construction process.

A good analogy from other domains of computing is the use of social security numbers for uniquely identifying persons in public databases. Another example is the inclusion in invoices of reference numbers issued by the receiver of the payments.

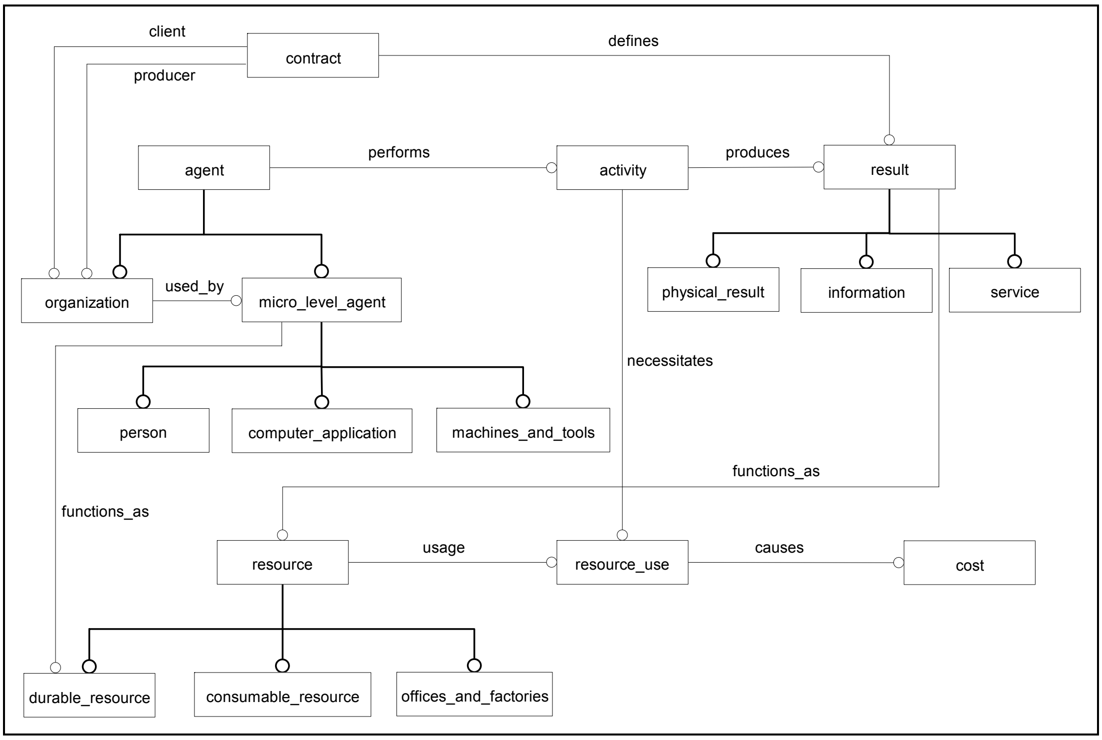

Figure 10. A model of highly generic different information categories used in construction computing applications.

This principle was tested in the particular case of bills of quantities in the fourth prototype and the results were quite convincing, especially for updating documents after design changes [III, p. 276]. Many problems with the propagation of changes in design information, which occur in the prevailing practise, could be avoided by enforcement of this principle.

6 DISCUSSION

6.1 CONTEXT OF THE ISSUES DEALT WITH

A useful way of summarising the research results presented in this thesis is to place the issues that received close attention during the study, within the layered framework presented earlier (table 4).

Table 4. The issues studied in this thesis in the context of the five-layer framework.

Model layer	Issues discussed in this thesis
Information modelling language	Choice of information modelling language.
Generic product description model	Modelling of type objects.
Building kernel model	Requirements for a building product model kernel. Extensions to the original RATAS five-level abstraction hierarchy. A conceptual model of the topology of spaces, space boundaries and enclosing structures. A generic model for construction process information.
Aspect models	Requirements for aspect models. Example prototypes for the energy conscious design of buildings and for quantity take off.
Application models	-

In the following, the results are discussed in the same order as they were presented in chapter 5, which hopefully facilitates the reading process.

6.2 REQUIREMENTS FOR BUILDING PRODUCT DATA MODELS

6.2.1 Capability for modelling any information type

This requirement is fairly basic to all product model work and it would be difficult to find researchers who would dispute it. It is, however, very important to state explicitly because of the diverse nature of the types of information dealt with in building design.

The demonstration of this principle in the four prototypes has had an important function in familiarising practitioners from the Finnish construction industry with the product model approach. In particular, the complete merger in the product model description of the information types to be found in briefs, drawings, building specifications and bills of quantity, has been important. Thus the emphasis was primarily on integrating information which is handled in separate documents for graphical, tabular/numerical and textual information. In the current generation of computer applications these documents are manipulated by separate software types such as CAD-systems, spreadsheets and word processors.

This requirement can also be compared with the 100 per cent principle of conceptual modelling theory [ISO 1985, p. 48]; "All relevant general static and dynamic aspects, i.e. all rules, laws, etc., of the universe of discourse should be described in the conceptual schema". The original formulation coincided with the ISO formulation, whereas the adopted one could be rephrased in such a way that the chosen modelling approach should be sufficiently expressive that it becomes possible to express all relevant propositions about the Universe of Discourse in the schema.

A different generic classification of information types needed in building product data models or computerised design descriptions is based on the three categories of *form, function and behaviour* [Luth et al. 1993]. Gero uses the terms function, structure and behaviour in connection with design prototypes [Gero 1990]. Such a classification is based more on the semantic meaning of information than on the classification into information categories typical of different existing construction document types. It is also more concerned with extending product models with information types which cannot easily be stored in traditional design documentation. The greatest interest in defining information models which also contain the behaviour of building components, is in structural engineering, a realm in which behaviour follows physical laws and where modelling behaviour is the core know-how of the profession.

6.2.2 Coverage of the information needs of different phases and disciplines

The requirements posed on building product data models in different phases of the construction process, have recently been discussed by Eastman

[Eastman 1993]. The main requirements that he discusses concern version control, integrity management, concurrency of operations and extendibility. Using the framework adopted in this paper, many of these information structures would fall into the generic product data level. Eastman concludes that the different phases of the design and construction process have differing requirements. Version control and integrity management are, for instance, more important in the design stages than during construction or facility management. He also clearly questions the paradigm of one monolithic building product data model.

A possible answer to Eastman's analysis is to use the kernel-aspect model architecture in such a way that issues such as version control and integrity management are dealt with outside the kernel and solved in separate aspect models. If the kernel is kept relatively simple, it can more easily cater to the requirements which are common to all phases of the design and construction process.

In the GARM model, information needed in the different phases of the construction project was modelled by different object classes for each phase (as planned, as designed, as constructed etc.) all of which were specialisations of a generic product definition entity [Gielingh 1988]. This classification was considered orthogonal to a number of other fundamental principles of classifying entities (e.g. dealing with the type object-occurrence object mechanism or with the building component classification). This contrasts slightly with the principle adopted in this project, according to which the same information structures in the kernel apply across life-cycle stages. In the opinion of this author, the information structures needed in different stages and disciplines of the construction process are more like subsets of the kernel. Subsets use the relevant information structures which correspond to the user views in different stages of the construction process.

The recently initiated STEP application protocol planning project for buildings, aims at defining the domains of a set of application protocols for the different technical systems and disciplines of the construction process [Wix et al. 1994] . The kernel-aspect framework is clearly a premise of this project, although the exact mechanisms for integrating the models on different layers are still not precisely defined.

6.2.3 Non-redundancy of information structures

The suggested reinterpretation of the non-redundancy principle (applies to the kernel, the combination of the kernel and any one aspect model, but not to the union of the aspect models) is in line with the recent development of the STEP standard. Earlier experiences with the IGES standard indicated that it is difficult for the developers of exchange standards or conceptual schemas to control how commercial software developers use them in their implementations. This is one of the primary reasons for the development of the application protocol feature of the STEP standard. Any data exchange using STEP adheres to some particular application protocol (i.e. for the exchange of data about prefabricated precast concrete elements) and not to the basic resource models of STEP itself. Application protocols reuse information structures from the resource models (and are consequently harmonised with the resource models) but there is no explicit requirement for application protocols to be harmonised between themselves. In view of the expected development of hundreds of application protocols, such a requirement would be impossible to adhere to.

6.2.4 Capability for supporting alternative presentation formats

Many other researchers have demonstrated the possibilities for producing different kinds of documents or computer screens from product model data. For the particular case of structural design, Howard and Howard demonstrated a mixed user interface mode for defining queries to a relational database [Howard and Howard 1988]. A research group from Pennsylvania State University have reported a hypermedia user interface to building product model data which is very similar to prototype number two of this study [Evt et al. 1992]. The Design++ system, an application development environment for knowledge-based design applications, provides a very good example of mixing geometrical modelling and object browser user interfaces to the same product model [Kulusjärvi 1990]. Design++ was originally developed for the design of industrial plants, but has recently also been used for developing building design applications. Figure 11 is an example taken from such an application.

6.2.5 Independence from restrictions of application software

The independence of the conceptual schema from the mechanisms of particular software paradigms is a principle which needs some clarification. Clearly a building product data model should not be tightly coupled with the syntax and data modelling capabilities of particular languages such as FORTRAN, SQL and C++. The experience gained in this study from using available commercial software verifies this conclusion. VTT's later OOCAD project, in which this author has not participated, has further highlighted the possibilities and limitations involved in using the data structuring capabilities of current commercial CAD technology for implementing product data model structures [Serén et al. 1993].

In the long run it is quite probable, however, that fully object-oriented application environments offer the best possibilities for implementing product model data models. A clear indication of this is the increasing use of C++ and object-oriented databases in prototype work.

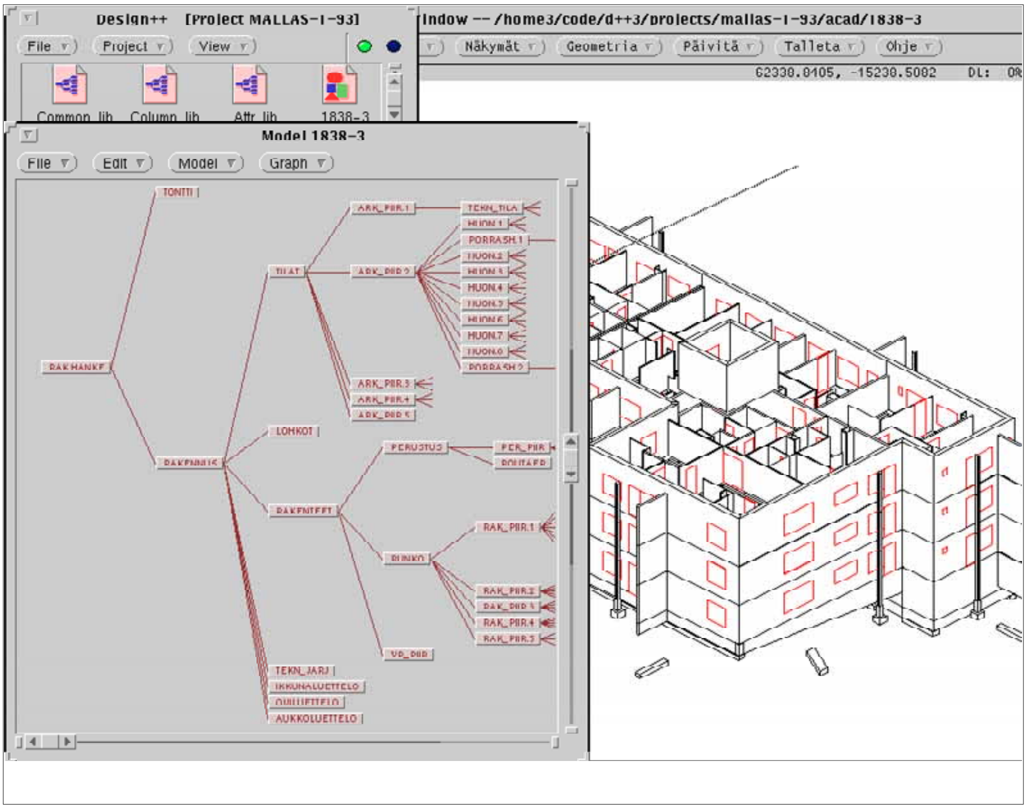

Figure 11. An example of a product model based application offering both geometric modelling and object browser user interfaces

6.3 INFORMATION STRUCTURES IN BUILDING PRODUCT DATA MODELS

6.3.1 Choice of information modelling language

The relatively cautious strategy suggested in this study for the choice of conceptual modelling language (relatively simple approach to start with, which later can be extended) was largely shaped by assumptions about what can realistically be handled in standardisation work. Other authors seem to favour semantically more powerful modelling languages, in particular containing capabilities for modelling constraints [Augenbroe et al. 1993], [Drach et al. 1994]. Some authors have argued for data models enabling certain operations with complex objects such as the structural model [Law and Barsalou 1989]. Eastman and Fereshetian have recently published an overview of a number of the most frequently used modelling approaches, comparing them with a set of criteria based on an analysis of the needs of information systems supporting the designers [Eastman and Fereshetian 1994].

A main dilemma seems to be the question of whether or not to include mechanisms to model behaviour in product data models. Many researchers feel that this would be important in order to provide mechanisms for the transfer of design knowledge between different applications [Dubois 1990], [Poyet et al. 1990]. A possible compromise would be to keep the conceptual modelling mechanisms of the kernel simple but to add more complex constructs such as derivation rules and more complex integrity constraints, in aspect models which need such features.

6.3.2 The abstraction hierarchy

This study suggests extending the original five generic levels of the RATAS framework model with objects for the total system formed by a site and the buildings on it, as well as for the site alone. This extension is relatively straightforward and follows from the extension of product model techniques to domains such as energy modelling and geotechnical engineering [Boulemia and Boissier 1993]. It will also facilitate the integration of product model data with the databases for urban infrastructure networks and with GIS-systems.

The extensive use of class structures for modelling properties, as suggested in this study, is facilitated by the use of particular types of conceptual modelling techniques. EXPRESS users who are defining STEP models are, for instance, recommended to avoid using primitive data types directly as the type of attributes. Thus STEP contains explicit definitions of classes such as *date*, which should be used as the data type of more specialised attributes such as *start date*, *end date*, *date of birth* etc. Such a principle leads to models which are difficult to implement in relational databases, but which can be handled in a relatively straightforward way using object-oriented programming languages.

Research related to the proposed strategy for modelling low-level attributes has been carried out at Stanford University, where a *primitive-composite* approach has been proposed for modelling information used in structural design applications [Howard et al. 1991]. The main difference to the proposal in this thesis is that in the primitive-composite approach, only primitive attribute type classes (such a as a form, a weight) are used for the actual data transfer between different applications. In the act of data transfer any composite level objects, such as a beam, would be split up into its constituent primitive objects only, from which the receiving application could assemble the type of composite object in which it is interested.

The information structures used for representing commonly used properties in the construction domain, would be an area extremely well suited for standardisation by the professional associations in the construction industry. Such work could also be pursued relatively independently of other product model work. Existing measurement conventions for particular building elements could, for instance, be included in such models.

6.3.3 The type object mechanism

Type objects have been studied extensively in a number of projects and are prominent features of some model proposals, such as GARM and OOCAD. The experience gained with the prototypes in this study, point out directions for further development of type objects.

The general solution proposed as a result of this experience (complex type objects and dynamic redefinition) is relatively difficult to implement in the short run with current software technology, but would offer the sort of functionality that designers would expect. Independently of this study, the implementation of dynamic redefinition and complex type objects has recently been tested in the OOCAD project, partly using conventional software technology (AutoCAD), partly using object-oriented technology (Actor) [Serén et al. 1993].

It should also be noted that there are new research directions in design theory and computer-aided design, which study techniques such as prototypes [Gero et al. 1988] or case-based design [Schmitt 1993]. Prototypes and cases are distantly related to type objects, especially to the sort of complex type objects advocated by this study.

6.3.4 Modelling spaces and enclosing structures

The proposed model for the topology of spaces, space boundaries and enclosing structures, was a synthesis of four models that had emerged from a number of relatively independent research projects in different countries. The author had participated in the elaboration of two of the original models. A particular aim of the synthesis model was to make a contribution towards the definition of the scope of a building kernel model, one part of which could be such a topology model.

This is a domain in which several models have been proposed [de Jong 1985], [Willems 1990], [Turner 1990b], [Turner 1990c], [Zamanian et al. 1992]. One feature which distinguishes this model from several of the other suggested models, is that it does not use geometrical or topological (vertex, edge, face etc.) information structures as direct representations of building elements. All relationships between objects are consequently purely functional (i.e. has holes, bounds, forms) and do not in any way involve the standard relationships between such geometrical entities. One of the benefits of the chosen strategy is that it should be easy to integrate different types of geometrical shape modelling methods to this model.

Other researchers have further developed the model as a part of their own modelling efforts. Dias discusses how to accommodate the structural and architectural view on building product model data [Dias 1993]. He proposes the use of three interface objects - surface, boundary and face - to link spaces and solids. A group at De Montfort University, England, has used the model as a basis for developing a prototype environmental design system [Lomas et al. 1995].

6.4 INTEGRATION WITH OTHER CONSTRUCTION PROCESS INFORMATION

The extension of building product data models to more comprehensive construction process information models, emerged as a new research topic about 1990. In the early phases, the efforts of this author [V], of Bart Luiten from TNO [Luiten and Tolman 1991] and of Thomas Froese from Stanford were made independently of each other. Luiten's model takes the GARM model as its starting point and extends it to include information about activities and resources as well [Luiten 1994]. Froese's model is the most detailed of the three [Froese 1992].

As a result of a workshop organised by VTT in 1992, a synthesis model of the earlier work, the Information Reference Model for Architecture, Engineering and Construction, was defined [Luiten et al. 1993]. The model is shown in figure 12.

The work on this model has later continued in the form of an electronic mail conference involving some 150 researchers world-wide [Froese et al. 1993].

The need for this kind of generic model has recently also been recognised by other parties such as the ISO committee which is defining a framework model for construction information [ISO 1993]. The need for integration in the direction of building product data models is receiving increasing attention also within the EDIFACT standardisation community [Neutebooms 1992].

The need for more formal approaches to integration is also starting to be recognised in other industries. Approaches of this kind have been named *enterprise integration modelling* [Petrie 1992]. This author has not come across any earlier mention of the unique object identifier principle in connection with computer integrated construction, although it is common principle in database theory. The introduction of such a usage would constitute quite a radical break with current practice where the same objects are often renamed by different participants in the design and construction process and where many objects often lack identifiers altogether. This principle should increase in importance as applications gain direct access to the databases of other applications.

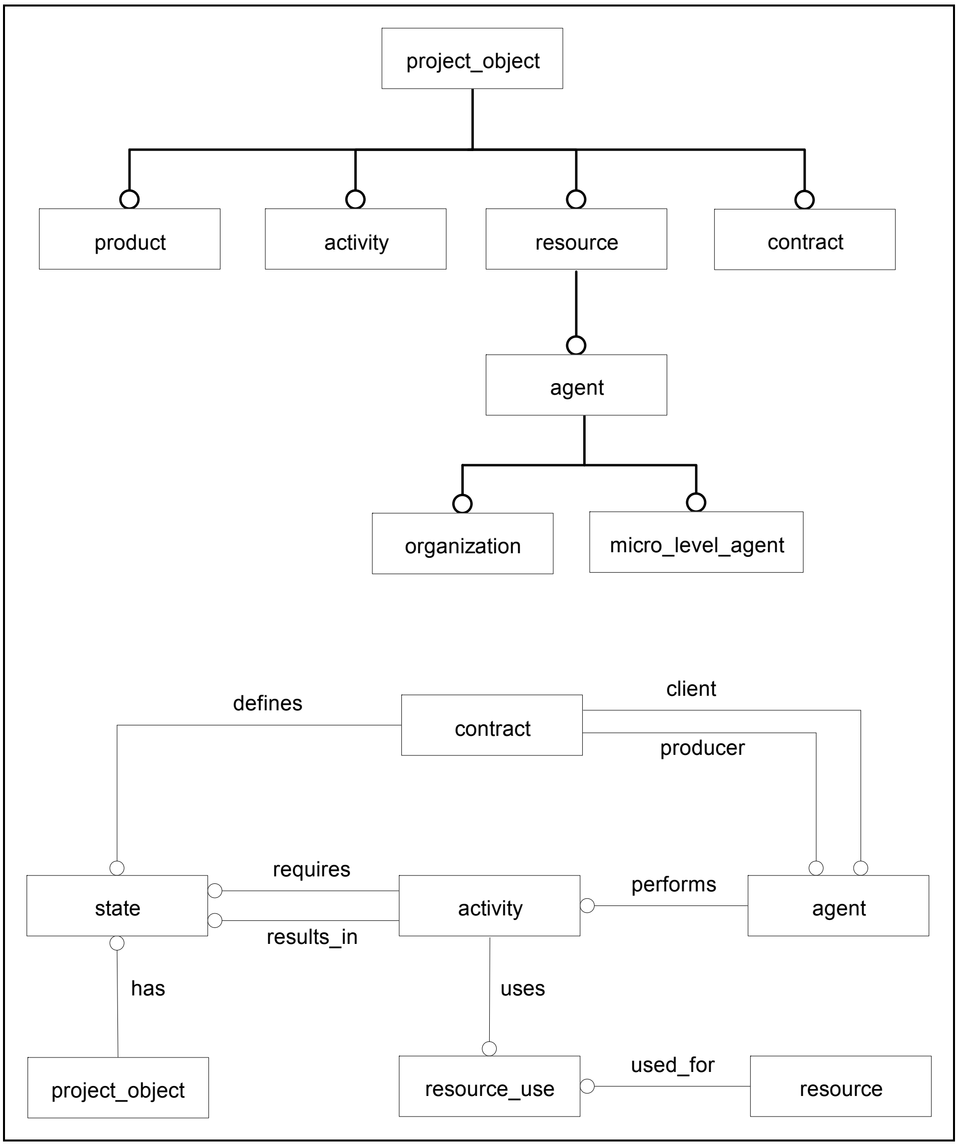

Figure 12. The Information Reference Model for AEC was defined in 1992 by a group of researchers interested in construction project models. The upper part of the figure shows the inheritance relationships, the lower part the functional dependencies [Luiten et al. 1993].

7 CONCLUSIONS

7.1 FULFILMENT OF THE AIMS OF THE STUDY

The aims of the study were mentioned earlier in section 2.3. In the following, an assessment is attempted of how well each of these aims was fulfilled.

- To elaborate on the criteria which a building product should fulfil and to study the fulfilment of these criteria using prototypes.

The criteria have been discussed at length in this summarising report. They were originally defined at the beginning of the study in 1988. During the course of the research some changes were made, due to feedback obtained from prototype work and theoretical analysis. In this final version, the criteria would seem to offer a tool which can be used to assess different conceptual schemas presented in the research literature, or as proposals for standardisation.

- To study in detail a number of critical generic information structures which are needed in building product data models.

The study concentrated on a number of information structures, which are discussed in sections 5 and 6. Results were obtained which, hopefully, can be of interest to other researchers and developers of commercial software. Some of the results indicate paths to be avoided (i.e. certain ways of implementing type objects) rather than workable solutions, whilst others offer schemas or embryos of schemas which could be reused or further developed.

- To test the applicability of the approach by building prototypes using a variety of different software techniques.

The study demonstrated the applicability, as well as the limitations, of relational databases, current generation CAD-systems and hypermedia software, alone and in combinations.

- To develop innovative user interfaces to design information structured according to the product data model approach.

This aspect of the study assumed increasing importance compared to the original aims. Certain types of user interfaces were developed (i.e. object browsers, room cards), which should be useful for the development of commercial software. The user interfaces proved extremely useful for demonstrating the whole approach to practitioners.

- To model a whole building in detail as a test case.

The study failed to achieve this aim. This was due to two factors. Firstly, the conceptual schemas which were developed in the project were not comprehensive enough. Secondly, the amount of work needed to collect, analyse and input the information, even in the sort of "dry run" originally envisaged, was greatly underestimated. A whole building was modelled in two of the prototypes, but only with the limited information related to narrow aspects (energy modelling, contractor's quantity management).

- To demonstrate the exchange of data between a building product model on the one hand, and technical analysis software and expert systems on the other.

This type of data exchange was demonstrated by a prototype for the case of energy analysis software. The use of product model data as input information for knowledge base systems was not attempted in this study, due to a lack of resources, but has been demonstrated by other researchers [de Waard 1992].

- To study how the information structures of the building product data model can be integrated with other information structures used in construction projects (i.e. for describing activities, resources).

A formalised conceptual schema on a highly generic level was developed as part of the study. This model has later been an important input for a model defined as international co-operation between a number of researchers [Luiten et al. 1993].

7.2 DIRECTIONS FOR FURTHER RESEARCH

The need for research concerning building product data models is by no means exhausted. There are still numerous technical issues which are unresolved. Nobody has so far suggested a fully detailed and widely accepted proposal for a building kernel model, not to mention a comprehensive building product data model. The validity of the kernel - aspect model approach still needs to be demonstrated. A major test will be its intended use as the basis for the standardisation activities carried out by the STEP building construction working group.

There are numerous technical issues for which adequate solutions are still needed. In the following, a few important ones are mentioned:

- Inclusion of information about the aims and intent of the designers.
- Controlling the versions of designs.
- Simultaneous alternative geometric representations of building parts.
- Integrity control on many different levels of ambition.
- Flexible extensions to the conceptual schemas defined by the users at the time of the creation of instances.

In addition to schema definition, testing of proposed conceptual schemas with full scale prototypes is urgently needed, as well as pilot use in real design and construction projects. So far, almost all of the prototype work reported in the literature has used only small subsets of the total information about a particular building. Many problems which arise from the complexity and volume of product models can only be studied with full scale models, including thousands of object classes and probably up to hundreds of thousands of instances.

Provided that product model based applications start to be used in real construction projects, a number of interesting research topics will emerge in the near future:

- Studies of information use and flows in the construction process.

Because product models help storing much of the information which currently is badly documented in the minds and personal notes of the participants in the design and construction process, the models should be of help for information tracking in empirical studies.

- The influence of the use of product models on the design process.

An often encountered hypothesis is that due to the use of product models, from which analysis software can easily obtain its input data, more alternatives will be evaluated in early design phases. This in turn should lead to better designs. An example would be more energy economic buildings as a result of simulation studies. This hypothesis needs to be verified.

All in all a major challenge to researchers in this field is the integration of product model research with design theory, which in the future ought to take account of the impact of product modelling technology on the design process.

THE AUTHOR'S CONTRIBUTION TO THE PUBLICATIONS

All the publications are based on research carried out by the author at VTT's laboratory of Urban Planning and Building Design. The first publication [I] builds partly on results of the earlier working group for building representation in database form of the RATAS phase II project. The author participated in the working group, which in addition consisted of eight CAD experts from industry and a research assistant from the author's laboratory. The role of the chairman of the committee, Matti Hannus, should be given particular mention.

Publications [II] and [III] were written as part of VTT's product model project. Some early results have also been included in publication [I]. The author had overall responsibility for the planning and execution of the project and carried out a large part of the theoretical work. He also set the strategies and aims for the prototypes, which were programmed by the junior researchers and research assistants participating in the project. Hannu Penttilä and Sven Hult should be mentioned in particular . For the theoretical work the contributions of Christer Finne and Kari Karstila were important. Finne participated in the project with a personal grant and completed a licentiate thesis on the architect's requirements for computer support during different stages in the design process [Finne 1992].

Publication [III] also contains some results from two additional projects which were direct spin-offs from VTT's product model project. They dealt with the use of the product model approach for energy-conscious design of buildings [Talonpoika et al. 1990] and management of quantities of construction materials [Penttilä et al. 1991]. In both these projects, the author's role was to plan them and supervise the junior researchers who carried out the bulk of the work.

Publication [IV] is based on theoretical work performed by the author as part of VTT's contribution to the COMBINE project [Augenbroe 1992], a large joint European study aimed at showing how the product model approach can be used for integrating energy analysis software with building design applications.

Publication [V] is directly based on the author's contribution to VTT's study concerning computer integrated construction, a project belonging to VTT's research programme for Information and Automation Systems in Construction [Björk et al. 1991a].

REFERENCES

Abb. 1993. Object oriented design using reference files. Abb publication 2E, AutoCAD users in building construction, Taastrup, Denmark.

Appleton. 1985. Information modelling language IDEF1X. Appleton Company Inc., USA.

Alexander, C. 1964. Notes on the synthesis of form. Harvard University Press. Cambridge, Massachusetts.

ATV 1994. Informationsteknolgi i byggeriet - FoU strategi. (Information technology in construction - a R&D strategy). The Danish Technical Research Council, Lyngby, Denmark.

Augenbroe, G. 1992. Integrated building performance evaluation in the early design stages. Building and Environment, vol. 27, no. 2, pp. 149-161.

Augenbroe, G., Chase, S., Rombouts, W. and Schipper, H. 1993. Information models of building design. In: Behesti, M. and Zreik, K. (eds) Advanced Technologies. Elsevier, Amsterdam. Pp. 355-364.

Autran, J. and Florenzano, M. 1985. CAO en architecture et SGBD. Le devis descriptiv de batiment: informatisation a l'aide d'un SGBD relationnel - limites et perspectives. GAMSAU, Université de Marseille, France.

Betts, M. 1992. How strategic is our use of information technology in the construction sector. International Journal of Construction Information Technology, vol. 1, no. 2, Pp. 79-97.

Betts, M., Mathur, K. and Ofori, G. 1989. Information technology and the construction industry of Singapore, a framework for a communications network. School of Building and Estate Management, National University of Singapore.

Beuke, K. and Ranglack, D. 1993. Computing with objects: What does industry hope to gain from it. In: Cohn, L. (ed), Proceedings of the Fifth International Conference on Computing in Civil and Building Engineering (ICCCBE). American Society for Civil Engineers, New York. Pp. 102-109.

Bijl, A., Stone, D. and Rosenthal, D. 1979. Integrated CAAD systems. EdCAAD Studies, University of Edinborough.

Bijl, A.. and Szalapaj, P. 1984. Computer literacy: designers' despair or hope. In: Proceedings of the Design Research Society Conference on the Role of the Designer. Bath, UK.

Björk, B.-C. 1987. Rakennusalan tietotekniikan tutkimus-, kehittämis- ja standardointitoiminta Suomessa vuosina 1981-86 (Research, development and standardisation activities concerning the use of information technology in Finland during the years 1981-86). Rakennusalan CAD-neuvottelukunta RACAD, Helsinki 1987.

Björk, B.-C. 1988. RATAS - A proposed Finnish building product model, Studies in Environmental Research, T6, Technical University of Helsinki, Department of Architecture, pp. 103-118., (also published in Acta Polytechnica Scandinavica, Civil Engineering and Building Construction Series, Ci 92, Helsinki 1989, pp. 97-106.)

Björk, B.-C. 1994a. The RATAS project - an example of co-operation between industry and research towards computer integrated construction. ASCE Journal of Computing in Civil Engineering, vol. 8, no. 4, pp. 401-419.

Björk, B.-C. 1994b. Conceptual models of product, project and document data; essential ingredients of CIC. In: Khozeimeh, Khalil (ed), Computing in Civil Engineering, proceedings of the First Congress held in conjunction with A/E/C systems '94, ASCE, New York, 1994, Vol. 1, pp. 980-987.

Björk, B.-C. and Keppo, J. 1984. Integroitu tietokoneavusteinen rakennussuunnittelu (Integrated computer-aided building design). VTT Tiedotteita 285, Technical Research Centre of Finland, Espoo.

Björk, B.-C., Hannus, M., Karstila, K., Koskela, L., Leppänen, P. and Sarja, A. 1991a. Tietokoneyhdennetty rakentaminen. (Computer-integrated Construction). VTT Tiedotteita 1319, Technical Research Centre of Finland, Espoo.

Björk, B.-C., Penttilä, H. and Karstila, K. 1991b. Rakennuksen tuotemallin teoria ja sen soveltaminen (The theory of the building product data model and its application, in Finnish). VTT Tiedotteita 1278, Technical Research Centre of Finland, Espoo.

Boman, M., Bubenko, J. and Johannesson, P. 1991. Conceptual modelling. Department of Computer and System sciences, Stockholm University. Second edition. Stockholm.

Booch, G. 1994. Object-oriented analysis and design with Applications. Second edition. The Benjamin Cummings Publishing Company, Redwood City, USA.

Boulemia, C. and Boissier, D. 1993. Applications intégrées autour d'une base de connaissances sur le sol. In: Behesti, M.and Zreik, K. (eds) Advanced Technologies, Elsevier, Amsterdam, pp. 165-174.

Brandon, P. and Betts, M. 1995. Integrated construction information. E&FN Spon, London.

Brodie, M., Myopoulos, J. and Schmidt, J. (eds) 1985. On Conceptual Modelling - Perspectives from Artificial Intelligence, Databases and Programming Languages. Springer-Verlag, New York.

BT 1995. Construct IT -bridging the gap. An information technology strategy for the UK construction industry. Final report. BT and Department of the Environment, London.

Chen, P. 1976. The entity-relationship model: towards a unified view of data. ACM Transactions on data base systems, vol. 1, no. 1, pp. 9 - 36.

Christiansson, P. and Karlsson, H., (eds). 1990. Conceptual modelling of buildings. CIB Seminar Proceedings, Publication 126, Working commissions W74 and W78. The Swedish Building Centre, Stockholm.

CIB 1972. CIB master lists for structuring documents relating to buildings, components, materials and services. CIB Report No. 18. Svensk Byggtjänst, Stockholm.

CIB 1986. A practice manual on the use of SfB. CIB Publication 55, An Foras Forbartha, Dublin.

Coad, P. and Yourdon, E. 1991. Object-oriented analysis. Prentice-Hall, 2nd. ed., Singapore.

Date, C. J. 1987. An introduction to database systems. Volume I. Fourth edition. Addison-Wesley. Reading, Massachusetts.

Delcambre, B. 1986. Towards an integrated C.A.D. for building projects. CIB.86 conference proceedings, National Bureau of Standards, Washington D. C., vol. 1, pp. 59 - 67.

Dias, W. 1993. Multi-disciplinary support in product models of buildings, paper submitted to the ASCE Journal of Computing in Civil Engineering, Dept. of Civil Engineering, University of Morituri, Sri Lanka.

Drach, A. , Heitz, S., and Langenegger, M. 1994. a + Prototypes Concept for individual support in tools for building design, ASCE Journal of Computing in Civil Engineering, vol.8, no. 4, pp. 489 - 506.

Dubois, A.-M. 1990. Data modelling for multi-actors design purpose. In: 2nd Finnish-French colloquium for Information Technology in Construction. VTT Symposium 118. Technical Research Centre of Finland, Espoo, Finland. pp. 39 - 48.

Dubois, A.-M., Escudié, J.-C. and Laret, L. 1992. COMBINE Integrated Data Model, Volume I - NIAM diagrams, Volume II - Dictionary, V.3.3, 920123, CSTB, Sophia Antipolis, France.

Dufau, J., Leininger, J.-P. and Miramond, M. 1987. Un système intègre d'evaluations technico-economiques d'avant projets de batiment organise autour d'une base de données alphanumeriques et graphiques. ARECDAO87 Conference Proceedings, ITEC, Barcelona, pp. 105-109.

Dupagne, Albert (ed) 1991. Computer Integrated Building. Strategic Final Report, Esprit II: Exploratory Action No 5604, CE Commission D. G. XIII.

Eastman, C. 1975. The scope of computer-aided building design. In: Eastman, C. (ed), Spatial synthesis in computer-aided building design, Applied Science Publishers Ltd, Barking, UK. Pp. 1-18.

Eastman, C. 1978. The representation of design problems and maintenance of their structure. In: Latcombe (ed), Application of AI and PR to CAD, North-Holland, Amsterdam. Pp. 335-337.

Eastman, C. 1992. A data model analysis of modularity and extensibility in building databases. Building and Environment, vol 27, 1992, no.2, pp. 135-148.

Eastman, Charles 1993. Life cycle requirements for building product models. In: Mathur, K., Betts, M. and Tham, K. (eds.) Management of Information Technology for Construction, World Scientific & Global Publication Services, Singapore. pp. 369-389.

Eastman, C., Chase, S. and Assal, H. 1993. System architecture for computer integration of design and construction knowledge. Automation in Construction, vol. 2, no. 2, pp. 95-107.

Eastman, C. and Fereshetian, N. 1994. Information models for use in product design: a comparison. Computer-Aided Design, Vol. 26, no. 7, pp. 551-572.

Enkovaara, E., Salmi, M. and Sarja, A. 1988. RATAS project - computer aided design for construction. Building Book Ltd, Helsinki.

Evt. S. K., Khayyal, A. and Sanvido, V. E. 1992. Representing building product information using hypermedia. ASCE Journal of Computing in Civil Engineering, Vol. 6, No. 1, pp. 3-18.

Fenves, S. J., Hendrickson, C., Maher, M., Flemming, U. and Schmitt, G. 1989. An integrated software environment for building design and construction. ARECDAO89 conference proceedings, ITEC, Barcelona. Pp. 71 - 83.

Finne, C. 1992. Arkitektarbetets väsen, datorstödet och planerings-processen, en analys och kravbeskrivning (The nature of the architect's work, computer support and the design process, an analysis and description of requirements) licentiate thesis, Helsinki University of Technology, Dept. of Architecture, Espoo.

Froese, T. 1992. Integrated computer-aided project management through standard object-oriented models. Doctoral dissertation, Dept. of Civil Engineering, Stanford University, California.

Froese, T., Cooper, G., Fischer, M., Oxman, R. (eds). 1993. IRMA-tica 93: An International E-Mail Conference on the IRMA model: an Information Reference Model for AEC. Transcript available in digital form through irma-l@jhunix.hcf.jhu.edu, University of British Columbia, Vancouver.

Fulton, J. 1995. Strategy for the integration of knowledge-based engineering data. working paper, ISO TC 184/SC 4. Stanford University. California.

Gero, J. S. 1990. Design prototypes: A knowledge representation schema for design. AI Magazine, Winter 1990, pp. 26-36.

Gero, J. S. 1994. Computational models of creative design processes. In: Dartnall, T. (ed). Artificial intelligence and creativity, Kluwer Academic Publishers, The Netherlands, pp. 269-281.

Gero, J. S., Maher, M. L. and Zwang, W. 1988. Chunking structural design knowledge as prototypes. In: Gero, J. S. edt., Artificial Intelligence in Engineering: Design, CPM/Elsevier, Amsterdam. Pp. 3 - 21.

Gielingh, W. 1988. General AEC reference model. ISO TC 184/SC4/WG1 doc. 3.2.2.1, TNO building and construction research, BI-88-150. Delft.

Gielingh, W. F. and Suhm, A. K. (eds.) 1993. IMPPACT reference model: an approach for integrated product and process modelling of discrete parts manufacturing. Springer Verlag, Darmstadt.

Glardon, C., Bedell, J. R., Hanrot, S. and Kohler, N. 1993. A building model to schedule and simulate the house refurbishment process. In: Behesti, M.and Zreik, K. (eds), Advanced Technologies, Elsevier, Amsterdam. Pp. 147-154.

GSD 1991. Synthèse des modeles conceptuels développés dans le cadre de la recherche batiment en France. Groupe Structuration de Données, Plan Construction et Architecture, Ministère de l´equipement, du logement, des transports et de l' espace, Paris.

Hannus, M. 1987. Object oriented product modelling in AEC applications. Proceedings of NBS/DATA Nordic Seminar Neste generations datasystem i byggebransjen, Norwegian Building Research Institute. Oslo. Pp. 100-115.

Hanrot, S. and Quintrand, P. 1988. Artificial intelligence and CAD in architecture, a knowledge-base. Proceedings of the 1st French-Finnish Seminar on Information Technology in Construction, CSTB, Sophia Antipolis, France.

Howard, C., Abdalla, J. and Phan, D. 1991. A primitive-composite approach for structural data modelling. Technical report no. 52. CIFE, Stanford University, California.

Howard, C. and Howard, C. 1988. User interfaces for structural engineering relational databases. Engineering with Computers, vol. 4, 1988, pp. 239 - 249.

Howard, H. C., Levitt, B. C., Paulson, B. C., Pohl, J. G. and Tatum, C. B. 1989. Computer integration: reducing fragmentation in the AEC industry. ASCE Journal of Computing in Civil Engineering, vol. 3, 1989, no. 1, pp. 18 - 32.

IOP-Bouw. 1989. BIM bouw informatie model version 3.1. (BIM building information model). IOP-Bouw, Rotterdam, 2 folders.

ISO. 1985. Concepts and terminology for the conceptual schema and the information base. ISO/DTR 9007 (TC 97). Published also as SIS technical report 311, Standardiseringskommissionen i Sverige, Stockholm.

ISO. 1992a. Draft International Standard DIS 10303 (STEP). TC 184. Starting in 1992 being published in several different parts by TC 184/SC 4.

ISO. 1992b. Draft International Standard DIS 10303 (STEP), Part 1: Overview and fundamental principles. TC 184 SC 4

ISO. 1992c. Draft International Standard DIS 10303 (STEP). Part 11: The EXPRESS Language Reference Manual.

ISO. 1992d. Draft International Standard DIS 10303 (STEP), Part 21: Implementation Methods; Clear Text Encoding of the Exchange Structure.

ISO. 1993. Classification of Information in the Construction Process. ISO working group TC59/ SC13 WG2, Technical Report N35E.

Johannesson, P. 1993. Schema integration, schema translation, and interoperability in federated information systems. Doctoral thesis. Stockholm University. Department of Computer & System Sciences.

de Jong, M. 1985. Spatial relational reference model. CAAD Futures Conference preproceedings, Technical University of Delft, The Netherlands.

Kalay, Y. (ed). 1987. Computability of design. John Wiley. New York.

Karstila, K. 1993. STEP/EXPRESS tools. Appendix in: Björk, B.-C. Byggproduktmodeller -Nuläge, Byggforskningsrådet R27:1993. Stockholm.

Kita, T., Kato, K., Wate, T., Takeda, M., Sakuma, H., Yamamoto, T. and Nambayashi, K. 1990. Loran-T: development and application of a newly integrated CAD system. Preproceedings of the CIB Working group 78 seminar Computer Integrated Construction, Architectural Institute of Tokyo.

Kulusjärvi, H. 1990. HVAC-engineering system description. In: 2nd Finnish-French colloquium for information technology in construction. VTT Symposium 118. Technical Research Centre of Finland, Espoo, Finland. Pp. 192 - 196.

Laitinen, J. 1992. The use of CIC in construction business. Preproceedings of the Joint International Workshop on Computer Integrated Construction and Computers and Building Standards, IRC, Montreal.

Langbecker, U. and Nowacki, H. 1994. A reference architecture for shipbuilding product models. In: Brodda, J. and Johansson, K. eds, ICCAS 1994, proceedings of the 8th international conference on computer applications in shipbuilding , Bremen. Pp. 6.125-6.139.

Lansdown, J. 1983. Expert systems: Their Impact on the construction industry. Report to RIBA Conference fund, London.

Lavakare, A. and Howard, C. 1989. Structural steel framing data model. Technical Report No. 012, CIFE, Stanford University, California.

Law, K. and Barsalou, T. 1989. Applying a semantic structural model for engineering design. Paper presented at the ASME Computers in Engineering Conference, Anaheim, California, Aug. 1989.

Law, K. and Jouaneh, M. 1986. Data modelling for building design. Proceedings of the Fourth Conference on Computing in Civil Engineering, ASCE, New York., pp. 21 - 36.

Lomas, K., Thornton, S. and Shulver, R. 1995. Environmental design of buildings. In: Brandon, P. and Betts, M. (eds). Integrated construction information. E&FN Spon, London. pp. 343-354.

Louden, K. 1993. Programming languages: principles and practice. PSW Publishing Company, Boston, Massachusetts.

Luiten, G. 1994. Computer aided design for construction in the building industry. Ph.D. thesis, Delft University of Technology.

Luiten, G., Froese, T., Björk, B-C., Cooper, G., Junge, R., Karstila, K. and Oxman, R. 1993. An information reference model for architecture, engineering and construction. In: Mathur, K., Betts, M. and Tham, K. (eds) Management of Information Technology for Construction, World Scientific & Global Publication Services, Singapore 1993, Pp. 391-406.

Luiten, B. and Tolman, F. 1991. A conceptual modelling approach to the development of integrated building project models and systems. 4th International Conference on Computing in Civil and Building Engineering, Tokyo, Japan, p. 7.

Luiten, B., Luijten B., Willems, P., Kuiper, P., Tolman, F. 1991. Development and implementation of multilayered project models. In: Mangin, J.-C., Kohler, N. and Brau, J. (eds), Computer Building Representation for Integration, Pre-proceedings of the second international workshop, Ecole polytechnique federale de Lausanne, Lausanne.

Lundequist, J. 1992. Projekteringsmetodikens teoretiska bakgrund (the theoretical background of design theory). Working paper, Institute of Design Methodology, Department of Architecture, Royal Institute of Technology, Stockholm.

Luth, G., McDonald, B. and Jain, D. 1993. Qualitative formulation of load paths through a functional description of structures. In: Cohn L. (ed.), Proceedings of the Fifth International Conference on Computing in Civil and Building Engineering (ICCCBE), American Society for Civil Engineers, New York. pp. 766 - 772.

Mangin, J.-C., Kohler, N. and Brau, J. (eds). 1991. Computer Building Representation for Integration. Pre-proceedings of the second international workshop, Ecole polytechnique federale de Lausanne, Lausanne, Switzerland.

Marca, D. A. and McGowan, C. L. 1987. SADT - structured analysis and design technique. McGraw-Hill, New York.

McIntosh, J. F. 1984. The application of the relational data model to computer-aided building design. Doctoral dissertation, University of Michigan, Ann Arbor, 1984.

Mitchell, W. 1977. Computer-aided architectural design. Petrocelli / Charter. New York.

Miyatake, Y., Yamazaki, Y. and Kangari, R. 1992. Current status of computer integrated construction in Shimizu Corporation. In: Preproceedings of the workshop on Frameworks for Computer-Integrated Construction, Björk, B.-C. and Hannus, M. (eds), Technical Research Centre of Finland, Espoo.

Molander, B. 1990. Vetenskapsfilosofi, Thales, Göteborg.

NBS. 1986. Initial graphics exchange specification (IGES), version 3.0, National Bureau of Standards, Washington D. C.

van Nederveen, G. A. and Tolman, F. P. 1992. Modelling multiple views on buildings. Automation in Construction, vol. 1, nr. 3 pp. 215-224.

Neutebooms, J. 1992. Bridging the gap between EDI and PDI. Preproceedings of the seminar Technical Data Interchange in Construction, CSTB, Paris.

Nijssen, G. and Halpin, T. 1989. Conceptual schema and relational database design, a fact oriented approach. Prentice Hall, London.

NRC 1986. Report from the 1985 workshop on advanced technology for building design and engineering. National Research Council, Building Research Board, Washington D.C.

NTNF 1984. Målstrategi 1985-1990. Datateknikk i bygg och anlegg. (Strategic plan 1985-1990. Information technology in construction). Norges Teknisk-Naturvetenskaplige forskningsråd, Oslo.

Owen, J. 1993. STEP, an introduction. Information Geometers, Winchester.

Owen, J. and Bloor, M. S. 1987. Neutral formats for product data exchange: the current situation. Computer-aided Design, vol. 19, no. 8, pp. 436-443.

Penttilä, H., Tiainen, P. and Hult, S. 1991. RATAS - building product model prototype, Quantity take off from a building product model representation of design data. ARECDAO 1991 conference proceedings, ITEC, Barcelona, (1991) pp. 83-90.

Petrie, C. (ed). 1992. Enterprise Integration modelling. Proceedings of the first international conference. MIT Press, Cambridge, Massachusetts.

Pohl, J., Myers, L., Chapman, A., Snyder, J., Chauvet, H., Cotton, J. Johnson, C. and Johnson, D. 1991. ICADS working model version 2 and future directions. Report CADRU-05-91., School of Architecture and Environmental Design, California Polytechnic State University, San Luis Opispo, California.

Port, S. 1984. Computer aided design for construction. Granada, London.

Poyet, P., Dubois, A.-M., Delcambre, B. 1990. Artificial intelligence software engineering in building engineering. Microcomputers in Civil Engineering, vol. 5, pp. 167-205.

Poyet, P., Debras, P, and Brisson, E. 1993. XPDI station. Working paper, CSTB, Sophia Antipolis, France.

RATAS-committee. 1988. Rakennuksen tuotemalli. (The building product model), Rakennuskirja Oy, Helsinki, Finland.

Rehak, D. R. and Howard, H. C. 1985. Interfacing expert systems with design databases in integrated CAD systems. Computer-Aided Design, vol. 17, 1985, no. 9, pp. 443-454.

Richens, P. 1978. The Oxsys system for the design of buildings. 3rd International Conference on Computers in Engineering and Building Design CAD78., Ipc science and technology press, UK, pp. 633 - 653.

Rumbaugh, J., Blaha, M., Premerlani, W. 1991. Object-Oriented Modelling and Design. Prentice-Hall.

Saarnivaara, V.-P. 1990. Information technology as a tool for industrialized construction systems and processes. In: 2nd Finnish-French colloquium for Information Technology in Construction. VTT Symposium 118. Technical Research Centre of Finland, Espoo. Pp. 10-30.

Sarja, A. and Leppänen, P. 1987. Preliminary models for the computer aided building design and construction management process. VTT Research Notes 606, Technical Research Centre of Finland, Espoo.

Schmitt, G. 1993. Case-based design and creativity. Automation in Construction, Vol. 2, No. 1, pp. 11-20.

Schenck, D. and Wilson, P. 1994. Information modeling the EXPRESS way. Oxford University Press, New York.

Serén, K.- J., Hannus, M., Karstila, K., Kihlman, M. and Pellosniemi, J. 1993. Object-oriented CAD tool implementations for the construction industry**.** VTT Research Notes 1460, Laboratory of Structural Engineering, Technical Research Centre of Finland, Espoo.

Simon, H. 1981. The sciences of the artificial. Second edition. MIT press. Cambridge, Massachusetts.

Sriram, D. 1986. Destiny: a model for integrated structural design. In: CAD86 conference proceedings, Butterworths, UK. Pp. 226-235.

Sriram, D., Logcher, R.D., Groleau, N. and Cherneff, J. 1989. DICE: an object oriented programming environment for cooperative engineering design. Technical Report IESL-89-03, Department of Civil Engineering, MIT, Harvard, Massachusetts.

Sterling, L., Shapiro, E. 1986. The art of Prolog. MIT Press, Cambridge, Massachusetts.

Svensson, K. 1991. Neutral building product model - “The KBS model”. Report T:123E, National Board of Public Building, Stockholm.

Talonpoika, R., Björk, B.-C., Penttilä, H., Heikkinen, L. and Kalliomäki, P. 1990. Olio-orientoituneet energiataloudelliset analyysi- ja suunnittelu-ohjelmistot (Object-oriented energy-economic analysis and design programs, in Finnish) Report 2, ETRR tutkimusohjelma, Technical Research Centre of Finland, Espoo.

Talo-80 group 1988. Rakentaminen 90. (Construction 90), Rakennustieto Oy, Hämeenlinna, Finland.

Tarandi, V. 1993. Object oriented communication with NICC, neutral intelligent CAD communication. In: Mathur, K., Betts, M and Tham, K. (eds) Management of Information Technology for Construction, World Scientific & Global Publication Services, Singapore 1993, pp. 517-527.

Turner, J. 1990a. AEC building systems model. ISO TC184/ SC4/ WG1, Doc. N363. Working paper.

Turner, J. 1990b. Spatial systems model. ISO TC184/ SC4/ WG1, Working paper.

Turner, J. 1990c. Building enclosure systems model. ISO TC184/ SC4/ WG1, Working paper.

Vanier, D. and Grabinsky, M. W. 1987. Integrated computer-aided building design systems. In: ARECDAO87 Conference Proceedings, ITEC, Barcelona. pp. 34-37.

de Vries, B. 1991. The minimal approach. In: Wagter, H. (ed), Preproceedings of the CIB W78 seminar Computer Integrated Future, University of Eindhoven.

de Waard, M. 1992. Computer-aided conformance checking. Doctoral dissertation, Technical University of Delft.

Wager, D. 1984. Expert systems and the construction industry. Construction Industry Computing Association, Cambridge, UK.

Watson, A.. 1995. To product models and beyond. In: Brandon, P. and Betts, M. (eds). Integrated construction information. E&FN Spon, London. pp.159-172.

Watt, D. 1990. Programming language concepts and paradigms. Prentice Hall, London.

Willems, P. 1990. A meta-topology for product modelling. In: Christiansson, P. and Karlsson, H. (eds): Conceptual modelling of buildings. CIB seminar proceedings, publication 126, working comissions W74 and W78. The Swedish Building Centre, Stockholm. Pp. 213-221.

Wilson, P. (ed) 1992. Processing tools for EXPRESS. ISO TC 184/SC4, working paper, Rensselaer Polytechnic Institute, USA.

Wix, J., Tolman, F., Poyet, P. and Björk, B.-C. 1994. Building Construction Core Model, Document N341, ISO TC184/SC4/WG3

Wright, R. N. 1988. Computer integrated construction. IABSE Proceedings P-123/88, IABSE Periodica 1. pp.17 - 25.

Wright, A.J., Lockley, S.R. and Wiltshire, T.J. 1992. Sharing data between application programs in building design, product models and object-oriented programming. Building and Environment, vol. 27, 1992, no. 2, pp. 163-171.

Yamazaki, Y. 1990. Integrated design and construction planning system for computer integrated construction. Preproceedings of the seminar Computer Integrated Construction, CIB Working Groups W78 and W74. Architectural Institute of Japan, Tokyo.

Zamanian, K., Fenves, S., Gursoz, E. L. 1992. Representing spatial abstractions of constructed facilities. Building and Environment, vol. 27, no. 2, pp. 221-230.

Appendices of this publication are not included in the PDF version.
Please order the printed version to get the complete publication
(http://www.vtt.fi/inf/pdf/)

Published by

Series title, number and report code of publication

VTT Publications 245
VTT–PUBS–245

Author(s)
Björk, Bo-Christer

Title
Requirements and information structures for building product data models

Abstract

The term computer-integrated construction (CIC) is often used to describe a future type of construction process characterised by the extensive use of information technology. The key to successful CIC is the comprehensive integration of currently isolated computing applications in different phases of the construction process. Among the several types of data exchange standards needed to support such integration, the standards for structuring the information describing buildings (building product data models) are particularly important. No fully operational building product data models have as yet been formally standardised either on the national or international level, but the topic has been a subject of intensive research during the last few years. Building product data model proposals are usually defined using object-oriented information modelling techniques.

The research which is presented in this summarising thesis was carried out primarily during the years 1988-92 at the Technical Research Centre of Finland. The report begins with a brief introduction to the general background of research concerning CIC and building product data models. Fundamental concepts of object orientation and product modelling are explained in a separate chapter. In order to position the author's research results, the "state of the art" in this research field is briefly reviewed.

The research results are presented against the background of a kernel-aspect model framework, in line with current thinking among several leading researchers in this field. The results can loosely be classified into three distinctive groups: a number of requirements which building product data models should fulfil; specific information structures in building product data models; and the integration of product models with other types of information used in the construction process. The specific information structures which were studied include the abstraction hierarchies used in building product data models, the type object mechanism and information structures needed for modelling spaces and enclosing objects.

The report ends with a discussion of the results, comparing them with the proposals and results of other researchers. Some directions for further research are also outlined.

Keywords
construction, product models, computers, integration, utilization, standardization, construction management, requirements, comparison, object programs, theses

Activity unit
VTT Building Technology, Lämpömiehenkuja 2, Espoo, P.O. Box 1800, FIN–02044 VTT

ISBN			Project number
951–38–4783–7 (soft back ed.) 951–38–6030–2 (URL:http://www.inf.vtt.fi/pdf/)			RTET93127
Date	**Language**	**Pages**	**Price**
December 1995; Electronic version February 2003	English	88 p. + app. 75 p.	D
Name of project		**Commissioned by**	
Series title and ISSN		**Sold by**	
VTT Publications 1235–0621 (soft back ed.) 1455–0849 (URL: http://www.inf.vtt.fi/pdf/)		VTT Information Service P.O.Box 2000, FIN–02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374	