Basic structure of a proposed building product model

B-C Björk

The basic principles enabling a building product model to be used by the application programs of the different participants in the building process have been defined in an industry-wide cooperation project in Finland. The model is conceptual, using concepts such as objects, attributes, and different types of relations between objects. The model is capable of containing all kinds of data describing a particular building. In current practice, these data are contained in drawings, specifications, bills of quantities, etc. At present the model is not fully elaborated, but it will be further developed in subdiscipline-specific projects, as well as being tested by prototypes.

building design, product models, abstraction hierarchy, object-oriented models

Integrated computer-aided building design has been a subject for research for nearly 20 years[1]. The basic theory was already quite well understood in the late 1970s[2], but due to limitations in computing technology, early prototype systems were rather cumbersome to use and imposed restrictions on possible design choices.

The first generation of CAD systems used in building design was, for practical reasons, primarily aimed at automating the production of drawings. Many popular systems have originally been developed for other industries or as general purpose systems. The adaption to the specific requirements of building design has often been made by the end users themselves.

The main tool for achieving a modest degree of integration between design disciplines has been overlay drafting, a technique originally developed for manual drafting[3], but now a feature of the data management systems of the majority of CAD systems. Layering is useful for many purposes, for instance for coordinating the work of the design team[4], but it is not an appropriate tool for achieving the ambitious aims of fully developed integrated CAD systems. At present there are efforts to standardize the use of layers, based on traditional building classification systems[5], but such standards will probably only provide interim solutions before the emergence of more powerful product models.

Recent advances in knowledge engineering and object-oriented programming techniques[6], as well as increasingly powerful hardware, are providing the technology for developing full-scale integrated building design systems. Before commercial systems emerge, a lot of work remains to be done on conceptual building data models. Conceptual models are models that tell us what kind of information is used to describe some aspect of reality and how such information is internally structured. They deal with the semantics of information as opposed to the syntax, which is dealt with in the physical models describing how to implement a given conceptual model as a computer database. The theory of conceptual models was developed in database theory in the 1970s. Closely related ideas were simultaneously being developed in artificial intelligence and programming language research[7].

Work on conceptual models for CAD/CAM/CIM has already started in connection with the development of standards for the transfer of CAD data between different systems. The emphasis in standardization is now clearly shifting from the transfer of geometric shape and drafting information (IGES) to the definition of product model standards (STEP and PDES)[8]. STEP will contain general purpose models applicable to any field of design and manufacture[9], as well as applications-oriented models[10,11].

At the same time, small prototype systems have been developed in several countries using knowledge engineering tools. Most of these prototypes have not tried to include all the design information describing a particular building, but have tried rather to concentrate on limited aspects of design (for instance, architectural[12], structural[13] or energy[14] aspects). Larger efforts on a national scale to develop full-scale conceptual models, which could evolve into national product model standards, are only starting[15–17].

In Finland a large number of building industry professional organizations have sponsored a national cooperative study (known as the RATAS project) in which the framework for the computer-integrated construction process of the future has been defined[18,19]. Among the results of the study is a basic structure for a product model to be used by the application programs of the different participants in the building process. The model uses such concepts as objects, attributes, and different types of relations between objects.

The model is capable of containing all kinds of building data, which is currently contained in documentation such as briefs, drawings, standard details, specifications, bills of quantities and calculation

Technical Research Centre of Finland, Laboratory of Urban Planning and Building Design, Itätuulenkuja 11, SF-21000 Espoo, Finland

 0010–4485/89/020071–08 $03.00 © 1989 Butterworth & Co (Publishers) Ltd

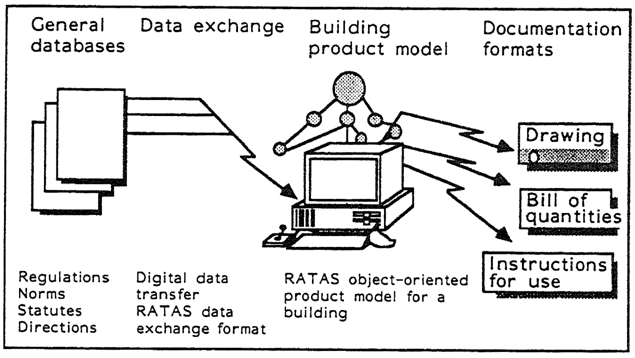

Figure 1. Schematic representation of the RATAS project, emphasizing the four main work groups

program output. At present the model is not comprehensive, but it will be further developed in subdiscipline-specific projects, as well as tested by prototypes. In the following, the main features of the RATAS model will be described. Figure 1 depicts the main interest areas in the RATAS project. The work was mainly carried out in the four groups shown in the figure: databases, data exchange, building product model and documentation formats.

MODEL REQUIREMENTS

The aim of the project was to define a conceptual model for structuring all data about a specific building, to be used in design, production and maintenance. Such a model should meet the following requirements.

Comprehensiveness

The model should contain all the information needed to construct and maintain a specific building. It should contain the information that is transmitted between the different design disciplines during the design process, as well as to and from the client and to the building regulatory authorities. Thus the model should cover information which in current practice is contained in drawings, specifications, bills of quantities, and similar documents.

Information contained in the conceptual model should describe parameters such as shape, location, relations between building parts, physical properties, materials used and price. Not all the information need physically be stored in a project database using the conceptual model. Some of the information, for instance data concerning detailed properties of prefabricated building parts, may be specified only as references to general databases maintained by building information services.

Coverage of all stages in construction process

The same building product model should be capable of expansion and alteration throughout the briefing – design – construction – maintenance process by all the different participants. In current practice, where documents are the only means for the different parties to communicate, the process is usually quite sharply divided into consecutive stages. Standardized types of design documents and drawings are associated to each stage. A building product model should also be capable of smooth continuous growth, as new design decisions are made and recorded in the product model. Definitions of different stages in the design process will implicitly be recorded in the 'recipes' for producing different types of output documents from the product model, but will not form a part of the model itself.

Avoiding redundant information

Each item of information should be defined only once in the model. Because of this, many problems that occur in manual practice are avoided. Nowadays the same data item occurs in many different documents and a revision of one should necessarily be made in all other documents where the same data item occurs. Due to this replication, contradictory information is a very common problem.

There is some question as to how far this approach of avoiding redundancy should be taken. In principle, the area of a room, for instance, can easily be calculated from geometric information concerning the walls that surround it. Should only this latter information be included in the database? Application programs operating on the database should then be able to deduct the area using rules. Alternatively, such generally applicable rules could be implemented in the conceptual model.

Output documents format and content

It should be possible to output the data contained in the model in very flexible ways as documents, which may be defined much later than the model itself. In this sense, content and format of output documents may be independent of the structure of the model. Figure 2 shows schematically how this development compares with a conventional design process. Traditionally, the design process has produced drawings and other kinds of documents, both for end users and as a way for designers to communicate with themselves and each other. The proliferation of computer-aided drafting systems has not changed this, but the introduction of an integrated project database (the building product model) should revolutionize design and design documentation practice.

Independence from software and hardware systems

The model should be a purely logical structure on the semantic level. It should be able to be implemented using several different programming and database techniques. In practice, however, it is evident that certain techniques are better suited to implementation than others.

The independence of the model also should mean that no concessions need be made in the definition of its structure to limitations posed by current hardware and CAD techniques. Such limitations are typical of several earlier attempts to define integrated CAD databases.

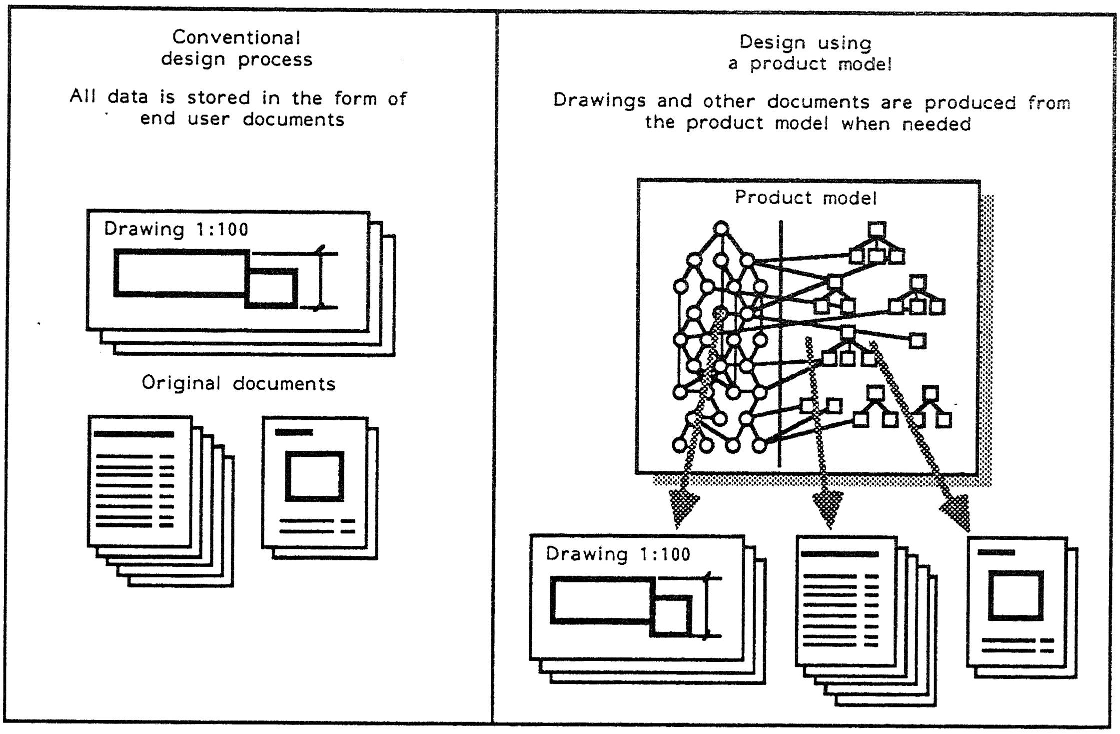

Figure 2. A schematic comparison of design and design documentation practice for the conventional design process and the process using a product model

RATAS MODEL

Objects

The RATAS model describes a building symbolically using objects (entities). Objects are collections of data that are related in some meaningful way. In the RATAS model, objects are formed through an analysis of the physical, and sometimes abstract, 'things' that can be found in a typical building. To each object, a number of attributes can be associated that describe the properties of the object in question.

The object description of a building is made up of objects and of a network of relations between these objects. Together these constitute a product model of the building. The description contains both geometrical and other data. In the RATAS work no formal method for defining classes of objects and especially the structure of relations between classes has been used. Such methods are for instance used in the STEP standardization work (IDEF1X[20], Express[21]).

Abstraction hierarchy

An abstraction hierarchy[22] is a tool that helps designers to deal with the subdivision of a building into meaningful systems and parts, and the internal relations between such objects. Objects from the higher levels in the abstraction hierarchy are especially useful in early design stages, for defining functional requirements and for making high-level design choices. Using such objects, it is also possible to describe data from the briefing and sketch design phases in the same product model as data from the detailed design phases. Objects from the lower abstraction levels are more 'physical' and often contain more attribute data, which is created in the detailed design phase.

The five abstraction levels chosen in the RATAS model are not the only possible ones, but seem appropriate for describing a building functionally. The levels are building, system, subsystem, part, and detail. These are illustrated in Figure 3. Alternative ways for organizing the abstraction hierarchy have been suggested in a number of prototype projects[12,23].

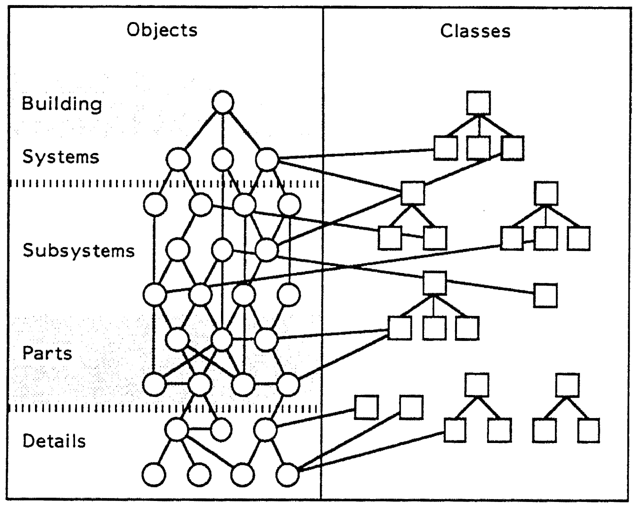

Figure 3. An illustration of the levels used in the RATAS system

Building level

There is only one building-level object per building. If a particular construction project involves the erection of several buildings, one building object should be used to describe each building. The building object contains attribute data about the site, the climate, the total size of the building, the construction cost or, for instance, the type of building.

System level

The system-level objects contain general information about the systems that together constitute the building. Using objects from this level, it is possible to collect information about the totality of lower level objects fulfilling certain functions in the building. All the spaces in a building form one system. All load-bearing and additional building components form a system. There are also several distribution systems in a modern building (heating, power, communication networks).

Subsystem level

Using subsystem objects, the designer can subdivide the above systems into functional parts (for instance, floor, hospital ward). Several partly overlapping subsystem objects can constitute the same system object through 'part of' relations. Objects from the part level are in turn part of the subsystem objects. An alternative term which could be used instead of 'subsystem' is 'group'.

Part level

The vast majority of objects belong to the part level. Part-level objects are mostly tangible physical objects such as building elements or technical devices. The part-level object space is a very important class of object. Typical attributes for all part-level objects are location and shape.

Data describing links between different part-level objects are also treated as objects. In a sense such objects describe relations between objects, but for many purposes it is useful to treat them as proper objects since there are attribute data attached to the relations.

Detail level

Many part-level objects may be further subdivided into detail-level objects (for instance, a window into its constituent parts). Sometimes this is done in detailed design, and sometimes it is left to the construction company. Manufacturers of prefabricated building parts do so in their own data systems. In the future, such information should typically be included in general access databases on building materials.

Attributes

Information about an object's properties (attributes) can be of many different types. In present-day documents, such attribute data can appear explicitly or implicitly in drawings, bills of quantites, and specifications. The most important attribute value types in the RATAS model are:

- Numeric values (for instance, geometrical data or prices)
- Text (strings of characters which have no meaning or internal structure but which can be transferred to different documents)
- Pictures (a bit map, or possibly analogue picture or video sequence that can be transferred to different design documents and viewed)
- Codes (strings of characters from a predefined set of allowable values, e.g. SfB codes[24])
- Lists (arrays of numeric values, texts, pictures, or codes)

An example of a set of attributes is shown in Figure 4.

Attribute domains

Each particular attribute has a predefined range of possible values. For many attributes the domain is as large as the range of all integers or all possible permutations of the characters in the alphabet. For some attributes, however, the domain is highly restricted. This is, by definition, true for classification codes. Such restrictions are very useful if the objects have to be grouped in ways which differ from the class hierarchies of the RATAS model for different kinds of analysis. In this way, classification systems, which are currently in use in the construction industry, can be integrated into the model.

A typical example, which is relevant in Finland, is the use of the attribute quality level for the building object. According to the classification used by the State Building Board in Finland, this attribute has four permissible values: I, II, III, or IV.

Building regulations and standards, as well as the client's demands, may impose limitations on the values of certain attributes. Other limitations are posed by what might be called 'good design practice', rules of thumb that represent accumulated professional expertise. Very often, limitations would not be straightforward, but would have to be expressed in the form of rules involving evaluation of the values of attributes of other objects in the product model. Such limitations are important in real design situations, but should be included in knowledge-based application programs which use the building model rather than in the building product model itself. The reason for this is that such rules may vary from one case to the other, necessitating changes to the software as building regulations change,

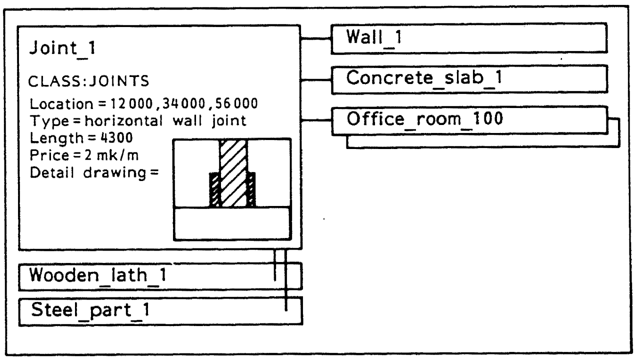

Figure 4. An example of a set of attributes of an object

or using the same software in different countries. It seems better to restrict the domain of the product model to the stable, invariant data structures at the core of computer-aided building design.

Attributes related to geometry

Crucial properties shared by almost all objects are location, orientation, and shape. The location might be specified in many different ways: in global or local coordinates, in relation to reference planes, or in relation to other objects. In the RATAS project, the detailed definition of location and shape attributes was not studied. One possible solution might be to allow several parallel location and shape descriptions in the model, possibly corresponding to the conceptual geometrical models underlying popular CAD systems, or to standards for data transfer such as IGES. In the future, it is possible that the evolving STEP standard might provide the structure for these attributes. In practical implementations of the RATAS model, these attributes have to be specified very strictly.

Classes

A class specifies the existence of attributes of the objects that belong to it. Each object is thus an instance of its generic class (the object has an 'is a' relation to its class) and contains specific values for its attributes. Each object belongs to at least one class.

Class hierarchies and inheritance

To model a building, in principle, a large number of independent classes could be defined, and for each class all required attributes could also be defined separately. However, this would be time-consuming, and would also lead to an obscure model structure, as many attributes are common to large groups of classes. If classes are arranged in meaningful hierarchies, common attributes can be defined for higher level classes. Lower level classes inherit the existence of such attributes from their higher level parent classes, in addition to which they can have attributes that are specific only to them. It would, for instance, be worthwhile to define a general class for all physical objects that would contain attributes for location, rotation, and mass. Thus, the objects belonging to such classes as structural components or doors would inherit the existence of these attributes from the class 'physical objects', rather than having them redefined for each class.

Relations

Above, a description has been given of how objects belong to classes, and how classes can be interrelated. These mechanisms are tools for determining which attributes are used for describing each specific object in the building product model. In order to describe the building as a product, data structures for describing how different objects are interrelated are also needed. Otherwise, the building description would only be a catalogue of the constituent parts of the building. Figure 5 shows an example of relations between objects.

Two types of relations are used for this purpose. The 'part of' relation specifies that a particular object belongs to a larger object. In principle, all other objects in the building model have a 'part of' relation to the building object itself. No other information is included in the 'part of' relation. Often, the 'part of' relation connects objects from different abstraction levels.

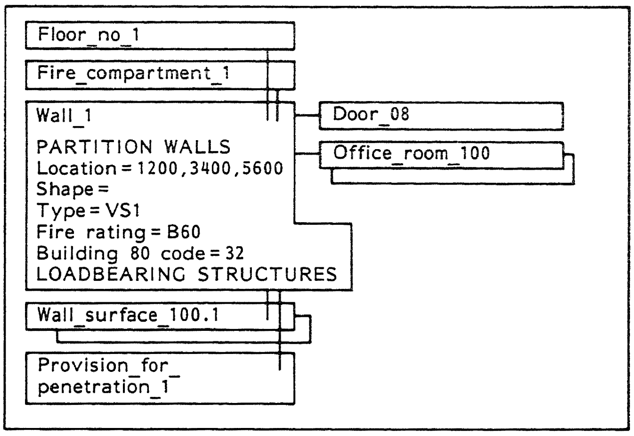

Figure 5. An example of relations between objects. The object Wall_1 belongs to the classes PARTITION WALLS and LOADBEARING STRUCTURES. It is part of the objects Floor_no_1 and Fire_compartment_1, and the objects Wall_surface_100.1 and Provision_for_penetration_1 are part of it. Furthermore, it is connected to Door_08 and Office_room_100

Another type of relation is the application-specific 'connected to' relation which was adopted in the RATAS project. It seems that this type of relation is especially useful for describing design artefacts. The 'connected to' relation is more typical for the lower abstraction levels (parts and details), and it usually connects objects from the same level. Since attribute data is often needed to describe the connection in question, these relations can be treated as objects of their own.

Views

Only in exceptional cases would a user need to consider the whole building at a time. More often, each user is interested in only specific object classes, relations, and attributes. In order to limit the data to be considered, views are used. Views can be attached to specific subdisciplines and specific design stages. Drawings and specifications are examples of documents produced from views of the total model. An example of the kind of document which will be available to the user is given in Figure 6.

FUTURE DIRECTIONS FOR RESEARCH

The results of the RATAS project are in no way revolutionary, since the basic theory has been in existence for some time, and a similar approach has been used in earlier prototype projects abroad. What is significant, however, is the prospective impact that the project will have on software development for the Finnish construction industry. Several of the leading

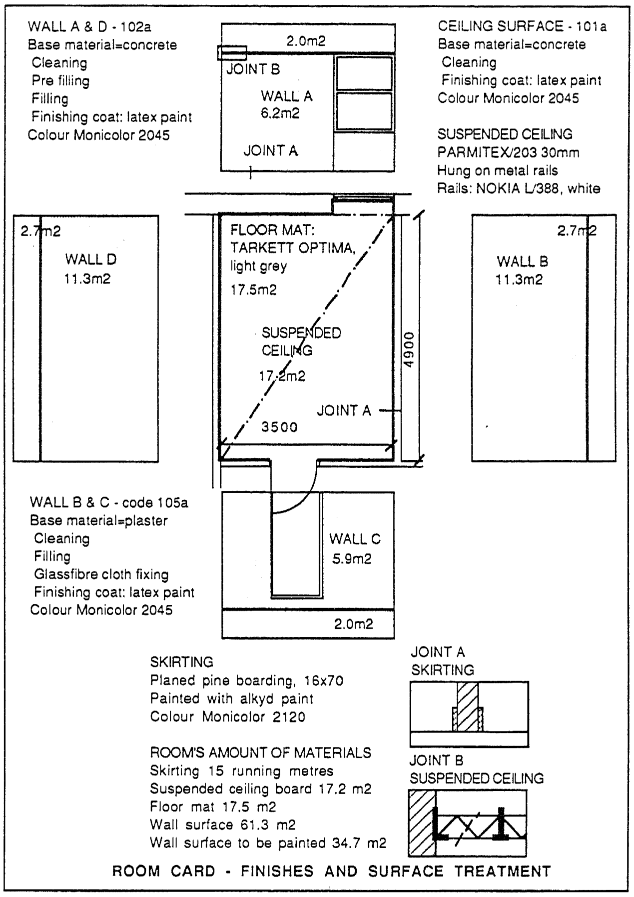

Figure 6. An example of a document produced from the information contained in the product model

EDP consultants working for the construction industry, as well as CAD system managers from design and construction firms, have participated in the project and have become familiar with the object-oriented approach. A number of follow-up projects have started or will start in the near future.

In connection with a three-year research program that will be carried out by the Technical Research Centre of Finland[25], an attempt will be made to build a full-scale prototype database using the RATAS model. In the first phase of the project, a prototype product model describing parts of an office building has been constructed using a commercial microcomputer-based relational database system. This model could be described as a bill of materials in computerized form, since it contains very little information about relations between objects and geometrical data. In later stages, the model will be extended with more attribute data and data describing relations. In addition, other prototypes modelling the same building will be developed using knowledge engineering tools. The prototypes will be tested as a source of input information for prototype expert systems for project management, energy analysis, and similar disciplines, which will be developed within the research program.

Apart from the straightforward prototype work, a number of fundamental theoretical issues will be studied in this project. The major issues are discussed below.

Basic data model

The structure of the fundamental data model underlying the conceptual building model needs to be rigourously defined. There are several possibilities with slight differences in the data structures provided (the relational model, the entity-relationship model and the frame approach are the most important ones). Researchers seeking more versatile tools for building CAD systems for architects have argued for using more complex, and consequently more expressive, data models[26]. There seems, however, to be a trade-off between the expressiveness of the data model and the potential for integration it offers. A simpler data model is easier to implement using a large variety of software tools, and seems to offer a better potential for providing a platform for an industry-wide product model. On top of such a basic model, application programs for different tasks might superimpose information with a more complex structure, for instance, design and regulations knowledge, as well as additional attribute data not defined in the common conceptual model. The only difference is that other applications programs have no access to such information, as it is not defined in the common standard.

The choice of an appropriate basic data model seems to be an important strategic issue that needs to be resolved in the near future. As yet, advanced computer applications have not been widely used in the construction industry, and it might be possible to influence software development, at least in a small country such as Finland. This requires, however, that at least a partial product model (containing major object classes and their most common attributes) should be defined quickly, before there is a large stock of existing CAD, expert and database systems in use in the industry. Provided that the chosen data model is upwards compatible with more complicated data models, it might be possible to extend the product model at a later stage, for instance, by standardizing the most common types of 'connected to' relations, without rendering software already being used obsolete.

Geometrical data

The description of shape and location data is an area for extensive research and development in the CAD community. No single optimum description has yet been found. It seems wise not to concentrate too much on this aspect in a small country with limited resources such as Finland, but to leave it to the market place to produce *de facto* or formal standards, which may later be incorporated in the conceptual building model. Within the product model, it is possible to allocate space for multiple descriptions of the same objects. It is also quite possible that the emerging STEP standard[9] might provide general purpose structures for geometric information, which could be used as part of an applications-oriented product model, such as the RATAS model.

Analysis of relations between objects

An interesting area for theoretical work is the analysis of the different types of 'connected to' relations needed to give a complete description of a building's structure. This is something which is highly industry-specific and which the building industry has to address itself.

A proposed PDES/STEP AEC Building Systems Model[27] has been the starting point for an interesting analysis of relations in a product model with methods similar to linguistics[28]. Four major categories of relations, active, dative, locative and partitive, were identified in the model. Interesting ideas have also been proposed in connection with the modelling of built facilities for robotics application[29]. One idea worth pursuing is to restrict relations to binary relations between two objects, and to include all sorts of relational data as facets of those single relations. The alternative is to allow several different relations connecting the same pairs of objects.

Product model in the design process

The interplay between the building product model and the design process also needs to be studied. Attempts to define conceptual models incorporating aspects of the design process include object classes such as functional units and technical solutions[9], which define further classifications of design information according to stages in the design process, as well as prototype-based design[30]. Commercial CAD systems also need to resolve problems of management of design versions, access rights, database integrity, and other issues.

The RATAS model at this stage makes no attempts to include aspects of the dynamic behaviour of design information, but might at a later stage be integrated with other models, for instance, those originating from the STEP work.

Concluding remarks

The RATAS system contains no new and revolutionary ideas from a scientific viewpoint. What makes the system rather unique is the amount of industry backing it has received. Provided that the necessary funding can be obtained for its further development, there is a reasonable chance that the system will be refined into a full standard taken into use in application software used in the Finnish construction industry, and that it will supercede the traditional classification systems now in use. It is hoped that similar efforts will be carried out in other countries, and that such work will eventually converge into a common building product model, as a part of the emerging STEP standard.

ACKNOWLEDGEMENTS

The author wishes to thank all the participants in the product model subtask of the RATAS project, in particular Mr Matti Hannus of Cadex Oy, who managed the project. His ten years of experience developing commercial CAD software counterbalanced the more theoretical background of the author in a very fruitful way. Mr Hannu Penttilä of the Technical Research Centre of Finland, who also participated in the project, deserves credit for making the illustrations, which have been essential for marketing the idea of 'objects' to people in industry.

REFERENCES

1 **Vanier, D and Grabinsky, M W** 'Integrated computer-aided building design systems' *ARECDAO87 Conf. Proc. ITEC* (Barcelona, Spain 1987) pp 34–37

2 **Bijl, A, Stone, D and Rosenthal, D** *Integrated CAAD systems* EdCAAD Studies, University of Edinburgh, UK (1979)

3 **Stitt, F A** *Systems graphics* McGraw-Hill, New York, NY, USA (1984)

4 **Silen, P** 'Use of integrated CAD in the building design process' *Proc. Int. Conf. Comput. Aided Architectural Des. – Developments in Education and Practice* (Singapore 28–30 May 1986)

5 **British Standards Institution** 'Guide for graphic representation by computer' *BS 1192 construction drawing practice draft part 5* BSI, London, UK (April 1988)

6 **Stefik, M and Bobrow, D G** 'Object oriented programming: themes and variations' *AI Magazine* Vol 6 No 2 (1986) pp 40–62

7 **Brodie, M, Myopoulos, J and Schmidt, J (eds)** *On conceptual modelling – perspectives from artificial intelligence, databases and programming languages* Springer, New York, NY, USA (1985)

8 **Owen, J and Bloor, M S** 'Neutral formats for product data exchange: the current situation' *Comput.-Aided Des.* Vol 19 No 8 (October 1987) pp 436–443

9 **Wilson, P R and Kennicott, P R** 'STEP/PDES testing draft; St Louis edition' *ISO TC 184/SC4/WG1 Doc. N 165* (September 1987)

10 **Gielingh, W F** 'General reference model for AEC product definition data' *Report BI-87-87 Version 3* TNO/IBBC, Delft, The Netherlands (September 1987)

11 **The Navy/Industry Digital Data Exchange Standards committee (NIDDESC)** 'Reference model for ship structural systems' *ISO TC184/SC4/WG1 Document N188 Version 2.0* International Standards Organization, Geneva, Switzerland (March 1988)

12 **Hanrot, S, Quintrand, P *et al*** 'A CAD system integrating architectural knowledge' (In French) *Proc. Fourth Int. Symp. Robotics and Artificial Intelligence in Building Construction* (Haifa, Israel 22–25 June 1987) Technion, Israel pp 430–441

13 **Sriram, D** 'Destiny: a model for integrated structural design' in *Knowledge engineering and computer modelling in CAD (Proc. CAD86)* Butterworths, UK (1986) pp 226–235

14 **Bourdeau, M** 'Conception thermique d'un batiment a l'àide du generateur de systeme expert SMECI' *ARECDAO87 Conf. Proc.* ITEC, Barcelona, Spain (1987) pp 72–76

15 **US Building Research Board** *Report from the 1985 workshop on advanced technology for building design and engineering* Building Research Board, National Research Council, Washington, DC, USA (1986)

16 **Terai, T** 'ACT – development of advanced construction technology systems utilizing advanced electronics' *Proc. NBS/DATA Nordic Seminar 'Datateknik i byggprocessen samarbete och gränsyta'* (Technical Research Centre of Finland, Espoo, Finland 1986) pp 77–86

17 **Le Quere, J** 'PROJIBAT – dossier de presentation du programme' (Centre Scientifique et Technique du Batiment, Paris, France 29 January 1986)

18 **Enkovaara, E, Salmi, M and Sarja, A** 'The RATAS system – for computer-aided building design' Building Information Institute, Helsinki, Finland (1988)

19 **Hannus, M** 'Object oriented product modelling in AEC applications' *Proc. NBS/DATA Nordic Seminar 'Neste generations datasystem i byggebransjen'* (Norwegian Building Research Institute, Oslo, Norway 1987) pp 100–115

20 **D Appleton Company Inc.** *Information modelling manual IDEF1 – extended* D Appleton Company Inc., Manhattan Beach, CA, USA (December 1985)

21 **Schenck, D** 'Express' *ISO TC184/SC4/WG1 Draft N210* (February 1988)

22 **Eastman, C M** 'The representation of design problems and maintenance of their structure' **Latcombe (ed.)** *Application of AI and PR to CAD, IFIPS Working Conf.* (Grenoble, France March 1978) North-Holland, Amsterdam, The Netherlands (1978) pp 335–337

23 **Law, K and Jouaneh, M** 'Data modelling for building design' *Proc. Fourth Conf. Computing in Civil Engineering* ASCE, New York, NY, USA (1986) pp 21–36

24 **CIB** 'A practice manual on the use of SfB' *CIB Publication 55, W74, Information Co-ordination in the Building Process* An Foras Forbartha, Dublin (1986)

25 **Technical Research Centre of Finland** 'Information and automation systems in construction' *Research programme* Technical Research Centre of Finland, Espoo, Finland (April 1988)

26 **Bijl, A and Szalapaj, P** 'Computer literacy: designers' despair or hope' *Proc. Design Research Society Conf. on the Role of the Designer* (Bath, UK 1984)

27 **Turner, J** 'AEC building systems model' *ISO TC 1184/SC4/WG1 Document 3.2.2.4 Working paper* ISO, Geneva, Switzerland (March 1988)

28 **Danner, W F** 'A global model for building-project information: analysis of conceptual structures' *Report NBSIR 88-3754* National Bureau of Standards, Washington, DC, USA (April 1988)

29 **Keirouz, W T, Rehak, D R and Oppenheim, I J** 'Development of an object oriented domain model for constructed facilities' **Sriram, D and Adeey, R A (eds)** *Artificial intelligence in engineering tools and techniques* Computational Mechanics Publications, Southampton, UK (1987) pp 259–271

30 **Gero, J, Maher, M L and Zhang, W** 'Chunking structural design knowledge as prototypes' *Working paper* The Architectural Computing Unit, University of Sydney, Sydney, Australia (January 1988)