A scenario for the development and implementation of a building product model standard*

BO-CHRISTER BJÖRK and HANNU PENTTILÄ

Technical Research Centre of Finland (VTT), Laboratory of Urban Planning and Building Design, Itätuulenkuja 11, SF-02100 Espoo, Finland

Product models offer a promising method for structuring data describing a building in the databases which will be used in the computer integrated construction process of the next century.

In order to define product model standards which completely describe modern buildings, many years of theoretical and experimental work are needed. In order to ensure industrial backing for the work needed to develop full standards, it seems prudent to organize such work on a step by step basis where partial standards can be agreed on and implemented rather quickly.

In addition to the international work being carried out within the ISO/STEP working group, it is possible to obtain results on a national basis, provided there is a large enough consensus within the construction industry on what ought to be done.

In Finland the basic principles of a building product model, the RATAS-model, have been defined. The model uses a data model consisting of such concepts as objects, attributes, relationships and it describes the building with a five-level abstraction hierarchy starting with the building object. At present a number of prototypes are being developed based on the RATAS-model. These prototypes are developed using different types of application software: relational databases, hypermedia, drafting systems etc., but they all utilize the same conceptual information structure.

1. INTRODUCTION

The integrated use of computers in construction necessitates the development of better methods and standards for building data management. Methods currently in use in computer-aided design systems are primarily graphics-oriented, and not adequate for the data retrieval needs of the application software of the future. In the computer-integrated construction[1] environment of the next century, typical application programs utilizing CAD data could be engineering calculation programs, expert systems for construction management, robot control programs and the building automation systems of intelligent buildings.

In the most advanced CAD-systems used today in building design, computer-aided drafting programs utilizing symbol libraries and layers have been interfaced with commercial relational database systems[2]. Using such systems it is possible to count the number of occurences of different types of symbols (e.g. doors, windows) as well as transfer attribute data connected to graphics elements for further processing with the tools offered by the data base systems. It is, however, difficult to integrate such systems with knowledge-based systems.

The most promising method for data management in the next generation of systems seems to be offered by standardized product models of buildings. Product models are conceptual structures specifying what kind of information is used to describe buildings and how such information is structured. The methodology used in such models is object-centred and based on advanced concepts from database, knowledge based systems and programming language theory. The central concepts are objects, attributes and relationships between objects.

A building product model should fulfill a number of criteria in order to provide the core of the computer integrated construction environment of the future[3].

Comprehensive	The model should be comprehensive. It should be capable of containing all types of data.
Cumulative	The model should cover the information created and needed during both the design, construction and operation of the building.
Non-redundant	The model should not contain redundant data.
Output independent	The structure and information content of output documents should be determined independently of the information structure of the product model itself.

* Paper presented at the symposium on Current Research and Development in Integrated Design, Construction and Facility Management, CIFE, Stanford University, California, March 28 and 29, 1989

Accepted June 1989. Discussion closes April 1990.

Hardware and software independently The product model standard should only specify what information is contained in the product model. It should not specify how this information is physically stored in files and records.

National and international work is presently under way to define product model standards for different application areas in industry. Internationally, the work carried out by the US PDES project and the International Standards Organization (the STEP standard[4]) is of importance. In Finland the basic structure of a national building product model (RATAS) has been defined.

AN IMPLEMENTATION SCENARIO

The possible success of any standard, provided that the standard is technically sound, is to a high degree dependent on how the development and implementation is organized and on securing industrial support. A crucial strategic issue in the implementation process of a product model is, for instance, choosing the level of detail of the standard. The speed and success of the implementation process may depend on this choice.

Product models are constructed using basic information structures defined in data models. Data models define the basic building blocks we use in the description of an information system. Typical concepts used in data models are: *entity*, *attribute*, *relationship*, *class*, *message*, *method*. Several data models have been proposed in data base theory and there are numerous minor variations[5].

A fairly simple and general data model is the entity-relationship (E-R) model[6]. Product models based on this approach are straightforward to implement in different types of software, for instance relational databases. More complicated data models are used in frame based knowledge engineering tools[7] and in object-oriented programming languages[8].

There are logic dependencies between different data models. A product model expressed using the tools of the E-R model can form a subset of a frame-based product model, but not vice versa. There thus seems to be a trade-off between the generality and the expressiveness of a product model. A simple data model is easier to implement in different kinds of software, and this gives the standard the early success which may be needed to obtain industrial support for its further development. A complicated data model on the other hand allows more possibilities for including knowledge in the product model, and consequently offers more information for the advanced knowledge-based systems of the future.

The most sensible path towards the development of a full building product model could consequently be to start by defining the basic entity types (such as *slab*, *window*, *room*) to be used in the product model, since these are contained in almost all possible data models. To a great extent these entity types can be found in already existing classification systems for construction. There are already examples of how classification systems can be utilized in defining national standards for CAD data management[9].

The second step would entail defining standard attribute types for each entity type and the third step adding standard types of relationships between objects. Attribute types may to some extent be found in standard specifications etc., but the classification of different kinds of relations is a totally new area, in which research is only starting[10]. These steps will demand much more work and should ideally be handled by the professional associations in the construction industry or by the organizations defining classification systems.

Further steps might include extensions using more complicated data models, including rules, methods etc; One area in which work is starting is the energy simulation area where there are proposals to standardize calculation methods and algorithms in order to ensure better compatibility between software[11]. Yet another direction is offered by the Global AEC Reference Model (GARM)[12] which is a part of the STEP standard. The GARM model offers entity types relating the information to different stages of the design and construction process.

Within each step there are further subdivisions according to sub-discipline, type of information (geometry, materials etc.). The benefits of this gradual definition and implementation process is that the results from each step are useful in their own right and that the resulting standards are upwardly compatible with later versions.

In this scenario research & development plays an important role. Theoretical work is needed for defining the

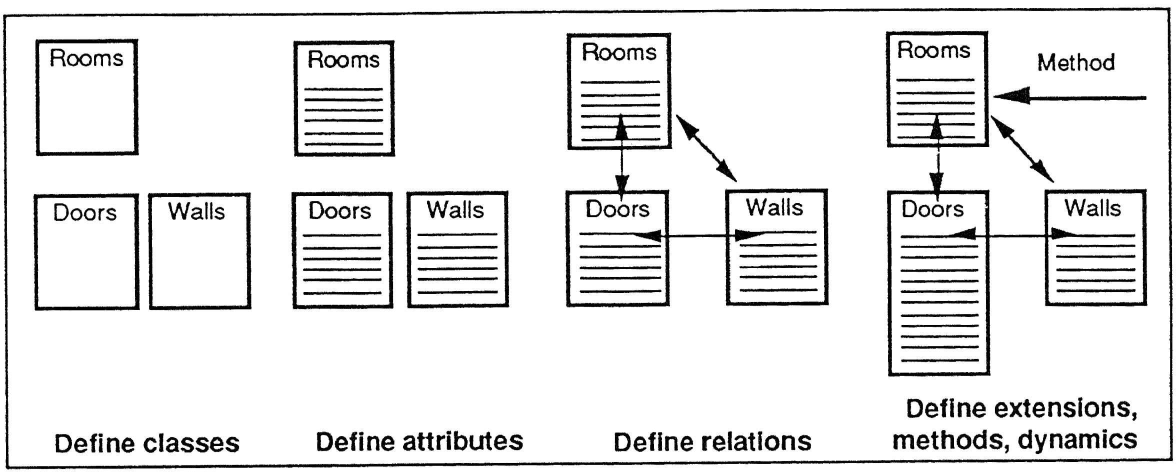

Figure 1. Gradual implementation steps in defining a product model standard

concepts used in the standard. Prototype work implementing selected aspects of the product model using different types of software is also urgently needed.

THE RATAS BUILDING PRODUCT MODEL

The aim of the RATAS-project has been to develop a national Finnish system for computer-aided design in the construction industry[13]. The system is meant for a situation in which the whole industry uses information technology on a large scale around the turn of the century.

The RATAS-project has been the largest and most comprehensive effort to develop construction computing in Finland. The project was carried out during the years 1987–88 with a total budget of 700,000 US dollars. Altogether 50–70 people representing 20–30 different organizations (research, design firms, contractors, prefabricated component manufacturers, commercial software development, government, etc.) participated in the project.

The kernel of the RATAS-system is the description of the building in computerized form – the Building Product Model. The other subprojects in the RATAS-project studied general databases, data exchange standards, and changes in design practice and documentation.

The model's structure

The RATAS-model describes a building symbolically using objects. A synonym term for "object" is "entity", but during the RATAS-project the term "object" was used. To each object one can associate a number of attributes, which describe the properties of the object in question.

The object description of a building is made up of objects and of a network of relationships between these objects. Together these constitute a product model of the building. From a formal viewpoint the RATAS-product model follows the entity-relationship data model quite closely.

So far, agreements have only been reached on the general principles of the product model (use of object-centred concepts, main abstraction hierarchy, etc.). The detailed definition of object classes, attributes and relationship types remains to be done.

An abstraction hierarchy[14] is a tool which helps designers to deal with the subdivision of a building into meaningful systems and parts and with 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 briefing and sketch data in the same product model as detailed data describing building components. Objects from the lower abstraction levels are more "physical" and often contain more attribute data.

The five abstraction levels chosen in the RATAS-model were: *building*, *system*, *subsystem*, *part* and *detail*.

Building The building object contains attributes about the site, the climate, the total size of the building, the construction cost or, for instance, the type of building.

System The system level objects contain general information about the systems that together constitute the building. All the spaces in a building form one system. All load-bearing building components too form one system. There are also several technical systems in a modern building (*heating*, *power*, *communication networks*, etc.).

Subsystem 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 objects through *part of* relationships. Objects from the part level are in turn part of the subsystem objects.

Part The vast majority of objects in the product model belong to the part level. Part level objects are most usually tangible physical objects such as building elements or technical devices. The part level object *space* is a very important class of objects.

Detail Many part level objects may be further subdivided into detail level objects (for instance a window into its constituent parts). In principle, the product model also covers the information structures on this level. In practice, such information can often reside in the general data bases provided by construction material manufacturers, etc. rather than in the database describing a particular building under design.

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

- Numeric values
- Text
- Pictures

Crucial properties shared by almost all objects are *location*, *orientation and shape*. In the RATAS-project the detailed definition of location and shape attributes was not studied. Other attributes types are more particular to each object class.

In order to describe the building as a product, we also need data structures for describing how different objects are interrelated. Two types of relationships are used for this purpose.

Part-of The *part-of* relationship specifies that a particular object belongs to a larger object. No other information is connected to the *part-of* relationship. Mostly the *part-of* relationship connects objects from different abstraction levels.

Connected-to Another type of relation is the application specific *connected to* relationship which was adopted in the RATAS-

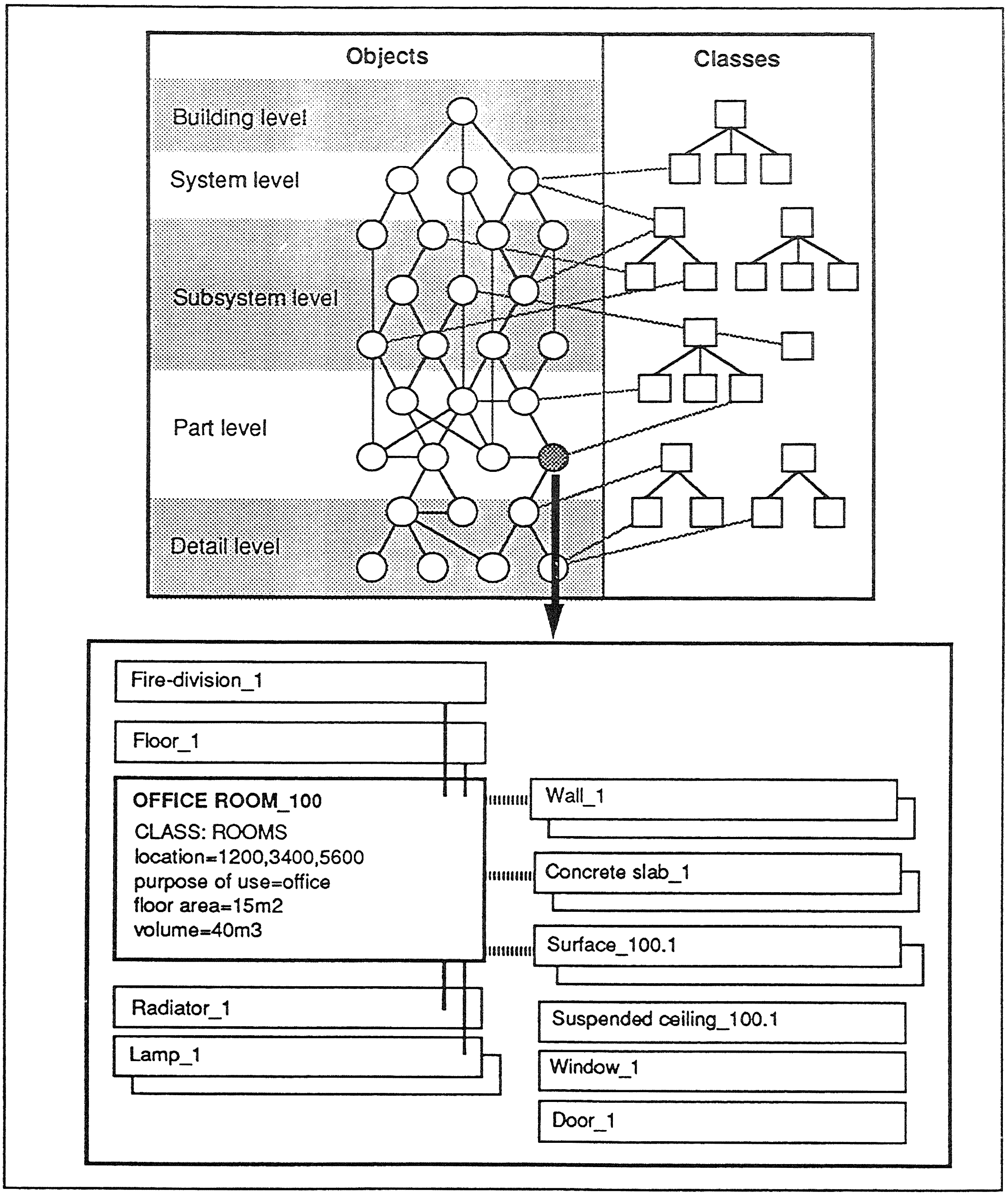

Figure 2. Overall structure of the RATAS Building Product Model

project. *Connected-to* relationships are more typical for the lower abstraction levels (parts and details), and usually connect objects from the same level. Since attribute type of data is often needed to describe the connection in question these relationship can in fact be treated as objects on their own.

PROTOTYPE WORK

In the spring of 1988 the Laboratory of Urban Planning and Building Design of the Technical Research Centre of Finland (VTT) began a three-year research programme called "Information and Automation Systems in Construction." The principal aim of the research programme is to develop the knowhow necessary for development work aiming at computer integrated construction[15]. In one of the subprojects of the programme, prototypes based on the RATAS-product model will be built. This prototype work will progress in parallel with other RATAS-influenced activities in Finland.

Already in the planning stages of the project some important restrictions on the work were agreed on. With the limited resources available (some five manyears), it is very important to concentrate only on a few key issues of strategic importance for the later definition of a full product model standard and for the development of commercial software based on such a standard. The most imporant issues and restrictions are listed below:

- The researchers should concentrate on the modelling of buildings, not on programming work. The prototyping work should, as far as possible, be done using state-of-the-art applications software (databases, CAD-systems, expert systems, etc.) and not in basic programming languages.
- Rather than trying to build one large prototype straight away, the project should build several gradually advancing prototypes, using different types of software, and later on integrate these. In this approach possible "blind alleys" would not lead to a total failure but add to the experiences to be learned from the project. So far (March 1989), two prototypes have been produced; one is in progress and one is in the planning stage.
- Little emphasis should be placed on the object-centred modelling of geometrical shape and location data. This area is huge and a lot of work is being done both by commercial CAD companies[16] and by the developers of the STEP standard. The main concern in VTT's project should be to make its own prototypes open, in the sense that they could later be integrated with other software, possibly adhering to other standards such as STEP.
- Another aspect which as far as possible has been excluded from the project is the interaction between the design process and the product model. In real CAD-systems using a product model approach, information has to be included concerning when and by whom information was created, who has the right to use and to change the information and what the status of the information is (functional requirement, chosen design solution, as built information, etc.). There is at present both theoretical[17] and standardization work being done in this area. Commercial CAD-system vendors will no doubt also develop their own solutions.

Prototype no.1: relational database

The use of relational databases for storing information describing buildings is by no means a novelty. Already in the early 1980's prototype CAD-systems for building design were being developed using relational database techniques.[18] More recently, commercial relational database systems have been interfaced with CAD-systems for storing non-geometric attribute data. Fairly straightforward applications for relational databases can be found in specifications writing and construction management[19]. The use of a relational database system for storing the complete data of a building product model is, however, much more complicated.

In order to make the exercise more realistic, a real building was chosen as a case. The researchers were able to use all the traditional documents describing this building (plan drawings, detail drawings, written specifications, bills of quantities, etc.). The tool chosen was a commercial relational database system (Oracle) running on a standard Intel 80386-based microcomputer.

The prototype deals with classes of objects, relationships between objects and non-geometrical attributes. It does not include the parameters necessary to generate two-dimensional or three-dimensional drawings from the data, but it does include areas, volumes, etc. It thus roughly covers the data usually included in building specifications or bills of quantities.

Rather than trying to refine the classes of the RATAS-model and their attributes in such a way as to cover most buildings, the research team concentrated on only that part of an ideal building product model which was needed to describe the building used as a case.

The building which was modelled is an extension to a four floor high office building with a total floor area of 5400 m^2. One floor of the building has been completely included in the database. In all, some 25 object classes have been modelled as tables, and the database includes some 5000 object instances. For most classes around 3 to 25 attribute types (average 8) have been defined but only for a part of these actual values have been input. All the data was entered manually by the project's research assistants.

Relationships between objects (in the entity-relationship model) can be implemented in different ways in relational databases. In this particular prototype, relationships between objects (*part-of* and *connected-to*) were modelled using separate tables containing the identifiers of the objects participating in the relationship, each in its own column. In order to make the relationship tables easy to comprehend, a naming convention was adopted whereby the name of a relationship table is formed by linking the names of the object classes participating in the relationship. Thus relationships between, for example, objects from the classes *columns* and *beams* are described in the table *columns–beams*. This way of modelling relationships is very general since it does not make any restrictions on the cardinality of the relationships (*one-to-one*, *one-to-many*, *many-to-one*, *many-to-many*).

The query in Fig. 4 finds all the surfaces, which are connected to a certain room-space. It is not trivial, since it picks the information from three separate data tables: rooms, surfaces and surfaces-rooms, which is a table connecting particular surfaces to particular rooms. The only key to make the selection used in this query is the room's unique identifier room-id.

The prototype has shown that it is possible to develop limited databases implementing the product model approach, and that such databases can already be very useful today[20]. Such databases could be used for structuring data, which today is presented in the form of building specifications and bills of quantities, in a more intelligent form. The job of putting the information into such a database from manual documents would not take more time than the traditional taking off of quantities, but the resulting database could prove very useful for the contractors who could use it for bidding, project management, materials procurement etc.

About hypermedia

During the last two years there has been an increased interest in the use of hypertext and hypermedia tech-

Object classes

BUILDINGS

SPATIAL SYSTEM
STRUCTURAL SYSTEM
HEATING SYSTEM
TELECOMMUNICATION SYSTEM
...
FLOORS
APARTMENTS
...
ROOMS
SURFACES
WINDOWS
DOORS
BEAMS
...

Relationships

FLOORS - ROOMS
ROOMS - ROOMS
ROOMS - SURFACES
ROOMS - WALLS
SLAB AREAS - SLABS
BEAMS - SLAB AREAS
BEAMS - SLABS
COLUMNS - COLUMNS
COLUMNS - BEAMS
WALLS - WINDOWS
WALLS - SURFACES
WALLS - WALLS
...

Figure 3. Examples of object-classes and relationships included in the relational database prototype

DATA TABLES

SURFACES-ROOMS

surf.	room
LP1	020
SP1	020
KP1	020
LP2	021

SURFACES

id.	type	code	explan.	col	area
LP1	floor	M1	Plastic.	323	15
SP1	wall	T2	Wallp...	-	18
KP1	ceiling	AK1	Met.....	-	15
LP2	floor	LK1	Nat....	221	55

ROOMS

id.	name	vol.	area
020	Office	45	15
021	Office	45	15
022	Wc	12	4
023	Corr.	150	55

SQL-QUERY

```
SELECT  Rooms. id, name, Surfaces . type, explan., code
FROM    Surfaces-Rooms, Surfaces, Rooms
WHERE   Rooms . id.=Surfaces-Rooms . room
AND     Surfaces . id.=Surfaces-Rooms . surf.
```

OUTPUT DOCUMENT

Output of a room-query				
020	Office	ceiling wall floor	Metal ceiling Wallpaper Plastic carpet	AK1 T2 M1
021	Office	ceiling wall floor	Metal ceiling AK1 Wallpaper Natural stone	AK1 T2 LK1

Figure 4. A room query from the relational database prototype

niques in database systems. These techniques also seem to offer interesting tools for presenting design descriptions.

Hypertext Hypertext is defined as nonlinear text, i.e. writing and reading in a "more human way", which means jumping from one thought to the other and connecting matters with appropriate links[21]. The basic idea is to allow the user to easily, quickly and associatively build these connections.

Hypermedia Hypermedia can, in addition, combine different information storage media: not just text but also graphics, pictures, animation, digitized speech, etc. Links and connections between different data items, no matter whether text, graphics or anything else, can be built just as with hypertext within an integrated computer-controlled environment.

Characteristic of hypermedia is the user interface. It is essential that the database is easy to access. For example, relational databases could be classified as hypermedia applications due to their flexible data structure, but their user interface is usually too rigid and clumsy. A basic concept in a hypermedia's user interface is a document window which shows one logical collection of information, such as a text document or a digital drawing. Links, relations or "jumpers" from windows to other windows should allow flexible and free combination of any data items.

All the information cumulated into the digitized data structure, forms a huge "semantic network", a collection of ideas and domains which cannot be fully described with any other tool but the hypermedia itself. There are several ways to browse through the information in the data structure.

- following the links between the data windows one at a time
- using a specific key query to find information in the network through database searches
- using a "browser" which shows the overall view of the data structure – "a map"

The impression the user gets of the data structure when browsing through the hypermedia is always very dependent on the actions at the moment of browsing. The hypermedia's data structure requires the user to participate actively in the process of gathering, creating, browsing through and analysing the information.

Hypertext, or more widely hypermedia, can be used in either collecting and browsing through existing data or on the other hand as "innovative problem exploration tools". An example of data browsing could be a library of references which combines a large written library into one literary environment to which one can refer freely within all the documents. Furthermore image libraries could be hypermedia solutions, for

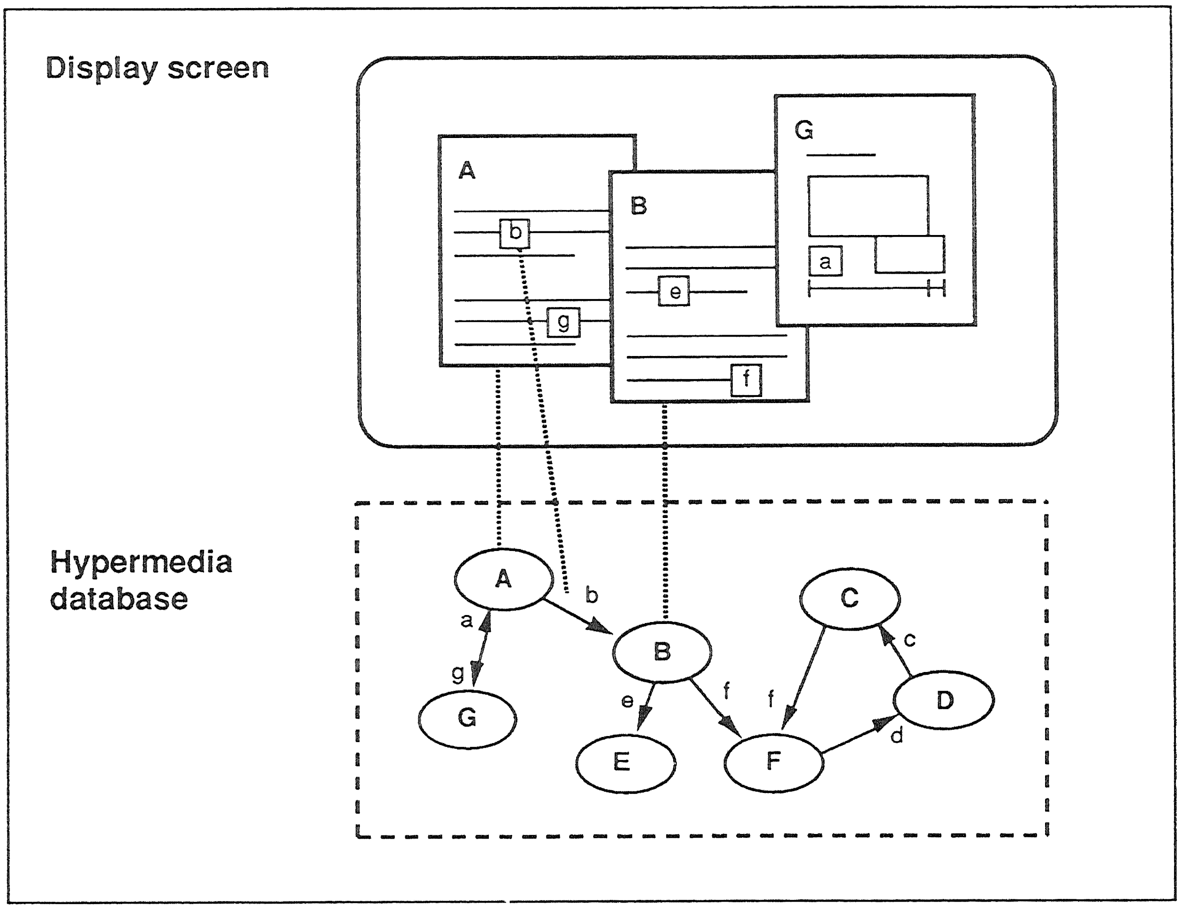

Figure 5. Example of a hypermedia's data structure[21]

example libraries of existing drawings with instant connections to other drawings describing the same objects.

Hypermedia based problem exploration tools could help to support outlining, thinking or sketch-like design and problem-solving especially in early process stages. Several commercial hypertext systems exist in various computing environments. Hypermedia has attracted much attention since the introduction of Apple's HyperCard software.

Prototype no. 2: hypermedia

The main purpose of this prototype is to provide a conceptual user interface to the building product model. The main emphasis in the prototype is in those stages of the building process where all the building information already exists in digitized form. The prototype shows one approach in organizing this information with a hypermedia tool.

The hypermedia prototype is much smaller than the relational database prototype since it only contains some 30 building objects belonging to 10 object classes. It was built in a Macintosh environment with HyperCard software which proved to be a quick and flexible tool for prototype development.

The prototype demonstrates new possibilities and practical tools for handling building information via different views. Tools to describe building objects, such as *rooms*, *walls*, *heating equipment*, etc. are conceptual, graphical and textual windows. Using several different windows simultaneously gives a good overall view of the whole product model structure. In addition to these basic types of user interfaces, it is also possible to extract information needed in the interfaces to other application programs.

Conceptual An object window shows all the attribute data describing one object and also all the relations the object has to other objects. One can jump from an object to all the related objects and in this way move within the network of building objects.

Graphical A drawing window shows graphical data related to the conceptual building object. Naturally, several drawings can be linked to one object. Existing drawings are connected to each other with logical links: a plan drawing of a window is connected to the section drawing and a detail drawing of the same window.

Textual A text window shows text documents dealing with the selected object allowing, for instance, access to all the written specifications.

Data transfer Provides facilities for exchanging product model data directly with different application programs and without a human interface. Queries can be programmed to pick specific data items in a specific form to be sent to a program, for instance a FEM analysis program. Another example could be a data retrieval query which picks a certain object's graphical attributes to form an IGES or STEP data transfer file.

The hypermedia prototype has, in a short time, already proved to be a valuable tool for demonstrating the con-

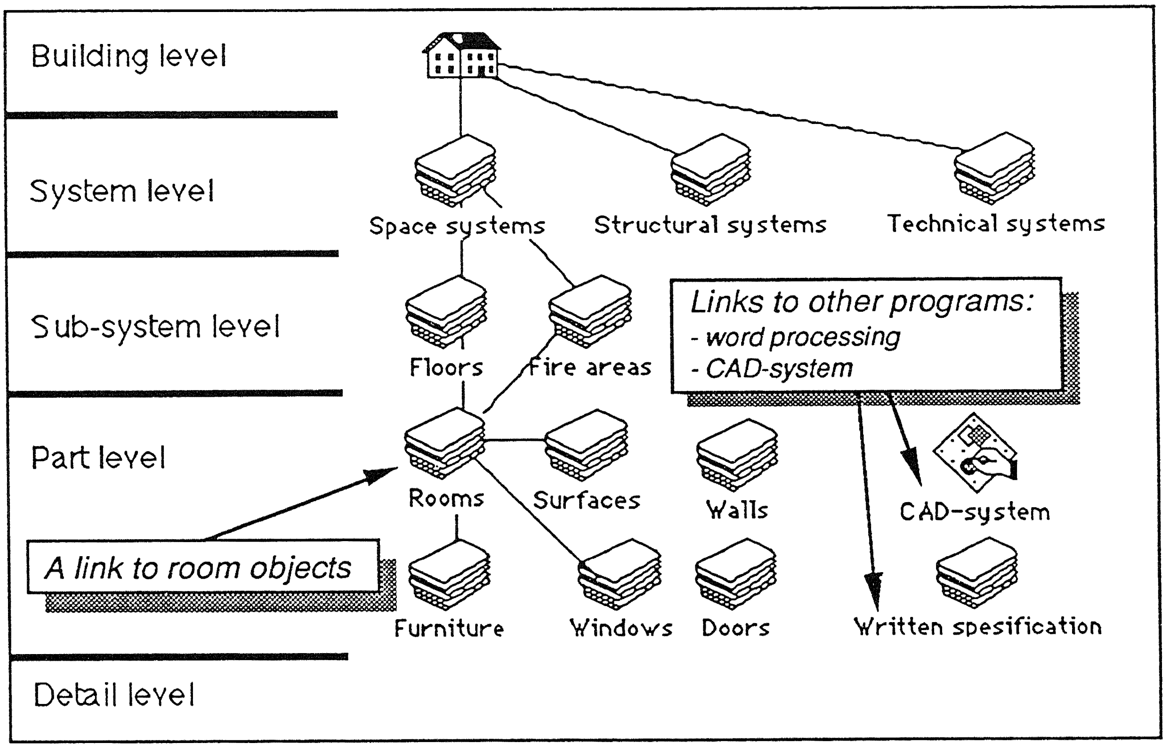

Figure 6. A conceptual overall view of the RATAS building product model. Several links exist in this window allowing the user to jump to other information in the model

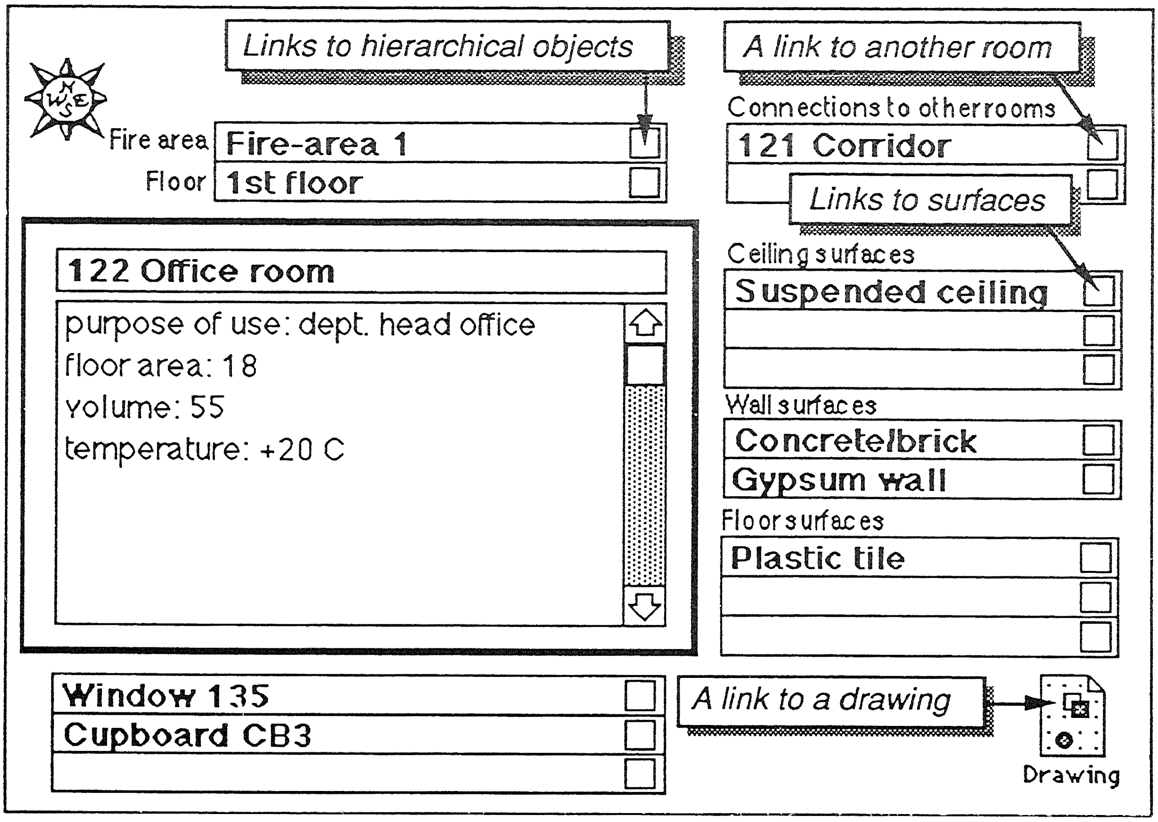

Figure 7. A conceptual view of a room object. Links exist to hierarchically related objects (part-of hierarchies), such as a fire-area. There also exist links to objects which are connected to the room with other relations (connected-to relations)

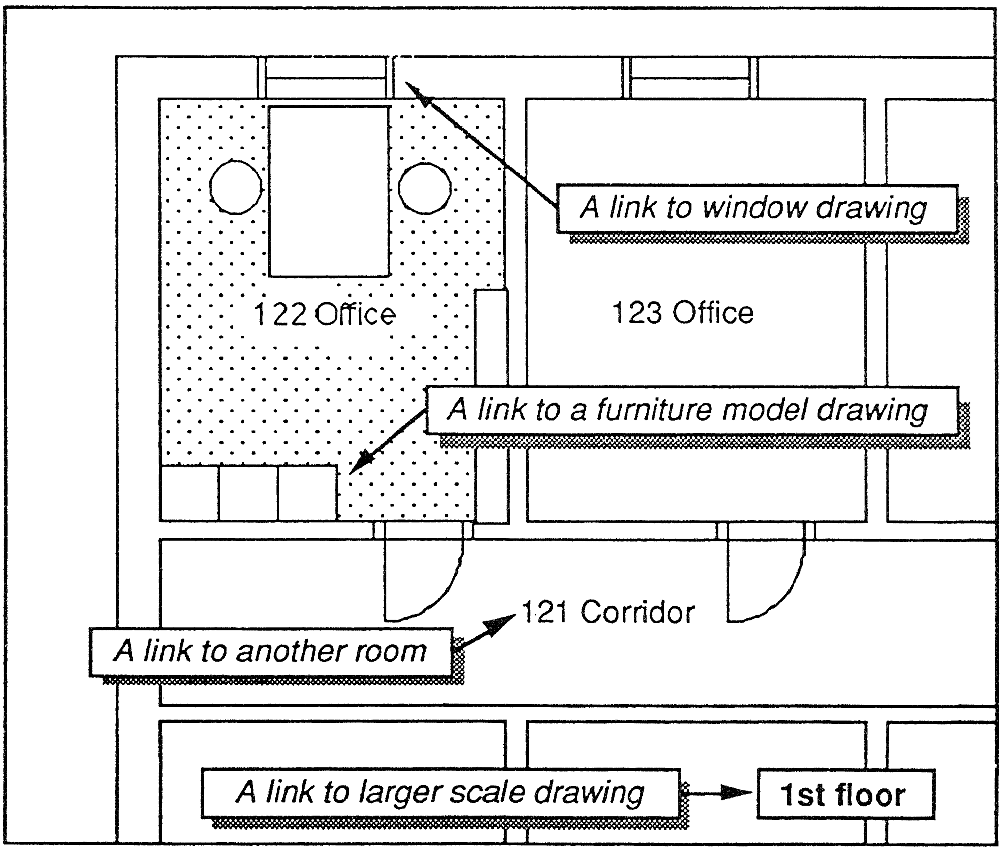

Figure 8. A graphical view of the above mentioned room object. A large scale plan drawing showing the overview. In this window, some links to other drawings or even 3-D model pictures are highlighted

cept of a building product model to people from the construction industry.

Further prototypes

The overall structure of a third prototype is at present being planned. The primary purpose of this prototype will be to demonstrate how the same conceptual product model can be implemented in different types of software to form an integrated system. Another aim is to include the attribute data necessary to produce 2-D graphical drawings from the database.

All the data included in the prototype will be found in a relational data base, following the guidelines set by the first prototype. The user interface to the data base will be through both a hypermedia program and a 2-D drafting program. The hypermedia access will resemble the hypermedia prototype described earlier.

The drafting system access will be based on the use of predefined parametrized symbols only. The use of graphical primitives such as lines, circles, points, etc. will not be allowed. Each symbol will correspond to a particular class of object (*wall*, *window*), and as soon as a symbol is positioned in a drawing, a line will be inserted in the corresponding object table in the relational database, including coordinate data. It is also important to note that one line in the relational database (representing a unique object) may correspond to several symbols to be used in different types of drawings. Drawings are not the main data store, they only provide one user interface access to the product model.

In addition to the product model itself, research is needed concerning application programs that communicate with the product model. In a separate project a prototype energy calculation program will be developed at the Technical Research Centre of Finland which interfaces with the product model and obtains its input data from it. The tool to be used for this application may be a frame-based expert system shell. This will, at the same time, neccessitate further definition of attributes in the product model which are needed for energy calculations (U-values, required termperatures, etc.).

During the next two years, work will also start to develop other object-centred application programs, for instance for cost estimation and construction management, which utilize the conceptual data structures of the product model.

CONCLUSIONS

The logical data structure for describing a building is very complicated and the quantity of information concerning one project is huge. A rough estimate of the number of objects one building includes might be in the scale of 20 000–100 000. Also the various data formats (drawings, alphanumerical specifications, time schedules, etc.) needed to describe a building and the

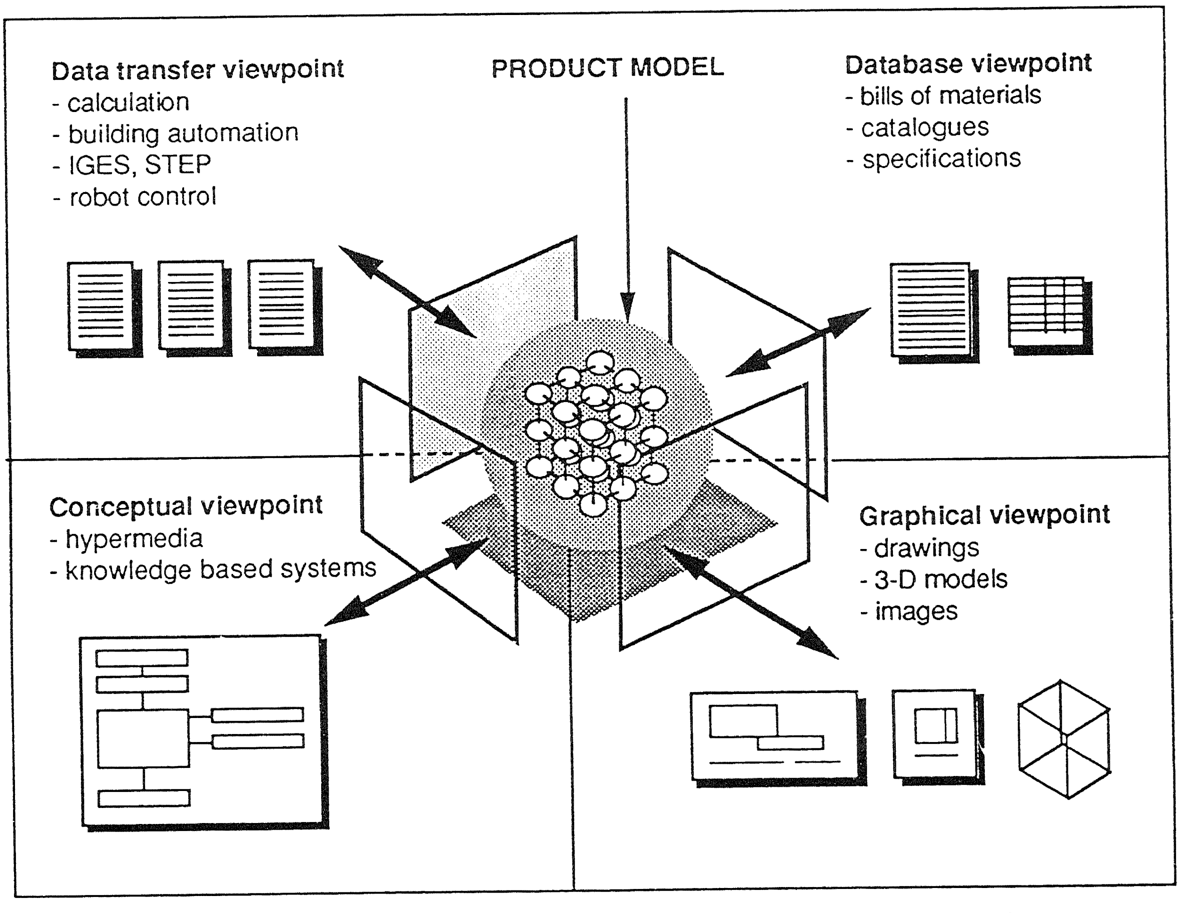

Figure 9. Product modelling is a very complicated subject. Different viewpoints point out different aspects of the model, but none of them completely describes the whole model

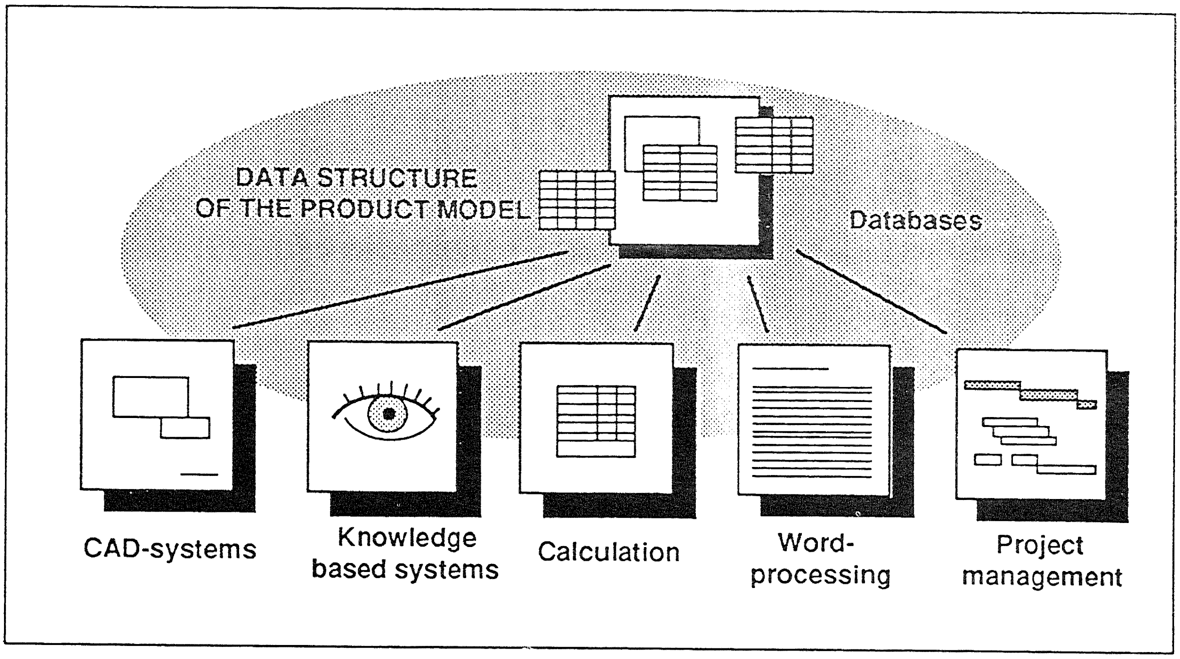

Figure 10. Building product model applications can be built with different software tools

construction process makes digital data manipulation in construction complicated. Characteristic too are the different viewpoints, each of which demand their own subset of the total block of information.

Building product modelling has in the prototype work proved to be a suitable way to structure building data. It offers a common data structure for each discipline concerned. It has also successfully been implemented into various software and hardware environments.

The small-step approach in developing the logical structure of the product model and in developing prototypes has been fruitful because it has created concrete results almost immediately. Pragmatic demonstration prototypes have quickly spread the idea of product modelling to key people involved in developing construction computing and classification systems. The small size of a country such as Finland has rendered this possible.

The researchers and developers involved in this work have long since realized that product modelling is an area where international cooperation is of vital importance. The work done in Finland, both in the RATAS-project and in the prototype project described above, has benefitted from the results of earlier theoretical work done in other countries and from work being done in the STEP project. Part of VTT's project, especially concentrating on fundamental, theoretical issues, will be carried out in cooperation with the National Institute for Standards and Technology (NIST), Washington D.C. Important international contacts are also provided by the working commission W78 ("Integrated CAD") of the International Council for Building Research Studies and Documentation (CIB).

REFERENCES

1 Wright, R. Computer Integrated Construction, *IABSE Proceedings* 1988 P-123/88, IABSE Periodica 1, 17–25

2 Penttilä, H. Data-Structures in Computer-aided Building Design, *Proceedings of the French-Finnish Symposium on Information Technology in Construction*, CSTB, Sophia Antipolis, France, Oct. 1988

3 Björk B. C. A proposed structure of a building product model, *Computer-Aided Design*, 1989 **22**(2) 71–78

4 Wilson, P. R., Kennicott, P. R. (eds.) ISO STEP Baseline Requirements Document (IPIM), ISO/TC184/SC4/WG1 Document N284, First Working Draft of STEP 1.0, Oct. 1988

5 Klein, H. K., Hirschheim, R. A. A comparative Framework of Data Modelling Paradigms and Approaches, *The Computer Journal*, 1987, **30**(1) 8–15

6 Chen P. The entity-relationship model: Towards a unified view of data, *ACM Transactions on data base systems* 1976 **1**(1) 9–36

7 Minsky, M. A Framework for Representing Knowledge, *In the Psychology of Computer Vision*, P. Winston (ed.), McGraw-Hill, New York, 1975, 217–277

8 Stefik, M., Bobrow, D. G. Object-Oriented Programming: Themes and Variations, *AI Magazine*, **6**(2) 40–62

9 Howard, R. Modelling Buildings and Classifying Data in Cad Systems, *Pre-proceedings of the seminar Conceptual Modelling of Buildings*, CIB working commissions W74 and W78, Lund, Sweden, Oct. 1988

10 Danner, W. F. A Global Model for Building-Project Information: Analysis of Conceptual Structures, National Bureau of Standards, Center for Building Technology, Report NBSIR 88-3754, Gaithersburg, Maryland, April 1988

11 Clarke, J. A. The Energy Kernel System, *Systems Simulation Conference*, University of Liege, Belgium, 1986

12 Giellingh, W. General AEC Reference Model, ISO TC 184/SC4/WG1 doc. 3.2.2.1, TNO report B1-88-150, Delft, Oct. 1988

13 Enkovaara, E., Salmi, M., Sarja, A. RATAS Project – Computer Aided Design for Construction, Building Book Ltd, Helsinki 1988

14 Eastman, C. M. The Representation of Design Problems and Maintenance of Their Structure, Latcombe (ed.), Application of AI and PR to CAD, *IFIPS Working Conference*, Grenoble, France, March 1978, North-Holland, Amsterdam, 335–337

15 Koskela, L. Research Programme for Information and Automation Systems in Construction, *Proceedings of the French-Finnish Symposium on Information Technology in Construction*, CSTB, Sophia Antipolis, France, Oct. 1988

16 Aish, R. Master Architect: an object based architectural design and production system, *Pre-proceedings of the seminar Conceptual Modelling of Buildings*, CIB working commissions W74 and W78, Lund, Sweden, Oct. 1988

17 Katz, R. H., Chang, E. Managing Change in a Computer-Aided Design Database, *Proceedings of the 13th Very Large Database Conference*, Brighton, UK, 1987, 455–462
18 Borkin, H. et al. Geometric modelling relational database system, Technical Report, Architectural Research Laboratory, The University of Michigan, Ann Arbor, USA, 1981
19 Betts, M. A Co-ordinated System of Information Retrieval for Building Contractor Tendering, Building Cost Modelling and Computers, Jan. 1987 Saiford, UK, 339–350
20 Björk, B. -C., Penttilä, H., Saarinen, H., Moisio, J., Finne, C., Nervola, M. A Prototype Building Product Model Using A Relational Data Base, *ARECDAO89 Conference*, ITEC, Barcelona, April 1989, 101–117
21 Conklin, J. Hypertext: An Introduction and Survey, *Computer* 1987, **20**(9), 17–41