PRODUCT MODELS OF BUILDINGS AND THEIR RELEVANCE TO BUILDING SIMULATION

Bo-Christer Björk, Senior Research Scientist
Technical Research Centre of Finland (VTT),
Laboratory of Urban Planning and Building Design,
Itätuulenkuja 11, SF-02110 Espoo, Finland
tel: +358-0-456 4562, fax: +358-0-464 174

Abstract

Computers are currently used for a large variety of tasks in building design and analysis. Among the basic software types used are 2-D draughting systems, 3-D modelling systems, spreadsheet and database programs, technical calculation and simulation software. One of the major drawbacks in today's situation is that almost every program uses a unique internal representation of the relevant data describing the building to be designed or analyzed. Consequently it is very difficult to exchange data directly between different programs.
A new approach to solving this integration problem is to standardize the representation of building data through the use of building product models. A product model structures both the geometrical information traditionally found in drawings and the information about materials, properties etc. found in specifications and bills of materials.
The techniques used for structuring product models come from database and knowledge-based systems theory. Basic concepts are entities or objects, attributes describing classes of objects and relationships between different objects.
ISO's Standard for the Exchange of Product Data (STEP) will hopefully provide a general product model standard applicable to all sorts of manufactured products, whether buildings, electronic circuits or machine parts. In Finland the basic principles of a product model for buildings (the RATAS-model) have been defined in a national industrywide co-operation project.

The widespread use of product model-based software should in the future enable simulation programs to get most of their input data directly from the CAD models of the designers. This, together with better user interfaces, should lower the threshold for using simulation software in everyday design work.

1. Introduction

Simulation has been an important application area for computing since the 1960's. In building design and analysis especially the thermal behaviour of buildings has been a subject for simulation.

Despite a quarter of a century of methodological development the simulation of the dynamic behaviour of a building is still today mostly a research and not an everyday design support activity. One of the major reasons for this state-of- affairs is the effort needed to input data into present-day simulation software. Almost all programs have unique internal formats for structuring the data describing the building being simulated. Despite the fact that many designers today produce drawings in digital form, the simulation programs cannot automatically access this data, due to incompatible data structures, and consequently the data has to be input manually.

Looking at the construction industry in a broader perspective the need for digital data exchange becomes even more pronounced as the number of computer applications in construction increases (computer-aided draughting, 3-D modelling, structural analysis, project management, expert systems for choice of materials, robot control et.c). Present-day standards for the exchange of CAD data, such as IGES, are geometry-oriented and inadequate, and new methods are urgently needed.

2. The Aim of product models

Since the middle of the 1970's, the period during which the data management methods used in many currently popular CAD-systems originated, many important developments have occured, which affect both CAD- and simulation software.[1] Alghoritmic FORTRAN programs and sequential or hierarchical files are no longer the only programming tools available. As we move on towards computer integrated manufacturing (CIM) systems, the databases, which many independent applications access, are becoming increasingly important in all fields of industry.

In recent years there has been a growing interest in the standardization of data structures for CIM systems. The emphasis in national and international standardization work is now clearly shifting from geometric shape and draughting information to the more comprehensive concept of product definition data. The concept is well explained by the following quote [2]:

"The term product data denotes the totality of data elements which completely define the product for all applications over its expected life cycle. Product data includes the geometry, topology, relationships, tolerances and features necessary to completely define a component part or an assembly of parts for the purposes of design, analysis, manufacture, test and inspection. Thus, no process data is included, but the product model is expected to enable the generation of all downstream process data."

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

- The model should be comprehensive. It should be capable of containing all kinds of data.
- The model should cover the information created during all the stages of the design and manufacturing process.
- The model should not contain redundant data.
- The structure and information content of output documents should be independent of the product model.
- 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 the computers.

3. An Overwiew of Database Theory

The development of product model based applications and databases will necessitate software and data management tools which differ from the tools used in current CAD systems, such as overlay draughting. Suitable new tools are now becoming available through the developments in knowledge engineering, object-oriented programming and data base techniques. The following brief overwiew of some fundamental concepts from database theory, knowledge-based systems and object-oriented programming is only meant to help the reader in understanding some of the ideas put forward in this paper.

Fundamental data structures for handling any information are called data models. A **data model** defines the basic information structures from which databases can be constructed. One of the most basic data models is the **entity-relationship model** (E-R-model).[4] The entity-relationship model is based on the concepts of entities and relationships between entities. Each entity is unique and corresponds to a physical or abstract object. The properties of an entity can be described with a set of attributes. Entities which have similar attributes can be grouped into entity sets. Similarily relationships of the same type can be grouped into relationship sets.

Another data model which has had a significant impact on the development of commercial database programs is the **relational model**, which is mathematically very well defined. In relational databases data is presented in the form of tables.

A more complicated data model which is very popular in knowledge engineering software is the **frame**. A frame can be viewed as a collection of related information about a topic. In addition to the kind of factual information which can be handled by the entity-relationship model frames can also contain functional or procedural information which can trigger actions as a result of the frame being accessed. The attribute of the E-R model is substituted by the slot, which is a very versatile information structure. Frames can also inherit the existence and values of slots from other frames, which is a very powerful mechanism for developers of complicated software systems. In addition to attribute type of information slots can contain other frames or methods, which execute actions. Frames are today increasingly used in sophisticated knowledge-based systems.

Frames have strongly influenced the development of a new computer programming paradigm called **object-oriented programming** [5]. The main difference between a frame and an object is the data encapsulation principle which applies to objects. This means that the information included within an object is only accessible indirectly by sending messages, which trigger methods within the object, responding to the message. The principle of data encapsulation has beneficial effects on the development and maintenance of complicated data programs. Object-oriented programming is beginning to influence the development of CAD-programs [6] and simulation software [7].

The analysis of full-scale databases can typically take place on three different levels: the conceptual, the physical and the user level.

The **conceptual level** contains a logical or semantic description of the data item types that are contained in a database and their interdependencies. It tells what information is contained in the database. The **physical level** deals with how the information structure described in the conceptual level has been implemented as physical records in database files. It deals with the exact format of the information stored in the database system.

On the **user level** part of the information in the database can be extracted from the database in the form of user views.

There are many conditions that must be fulfilled before different application programs in the construction industry can exchange data. First of all the physical data formats must be compatible. Compatibility on the physical level is fortunately relatively simple to handle via conversion programs between different physical data formats. More important in the long run and more difficult to achieve is compatibility on the conceptual level. Different applications should "talk about the same things". The purpose of product models is essentially to ensure this.

4. The Standard for the Exchange of Product Data

Work on standards for product models is currently centered in the joint efforts of PDES and STEP. PDES (Product Data Exchange Specification) is an initiative of the US IGES committee. STEP (Standard for the Exchange of Product Data) is being defined by the International Standards Organisation ISO. These two efforts are highly coordinated and will eventually result in one single standard.

STEP covers the description of all kinds of manufactured products throughout their lifetime. In addition to defining common entities related to any products STEP will include reference models for different application areas.

Since the work on STEP is done by so many individuals great care has been taken in choosing appropriate work methods. The definition of all conceptual models is done using a formal data definition language called EXPRESS[8], which contains elements from many of the fundamental data models described above. This means that it is easy to integrate the work done in different subcommittees into one large coherent model.

STEP follows a three layer architecture:

The **logical layer** of STEP corresponds to the conceptual model of data as defined above. The logical layer contains definitions of entity types which are common to all application areas (basic geometrical entities, tolerances et.c). The geometric entities defined in STEP will include the entities needed in most commonly used geometric modelling systems, such as wireframe, surface, boundary and constructive solid representation.

The **physical layer** describes the representation of the product data in a sequential STEP-file. The actual exchange of data between different applications will take place using this file structure.

The **application level** will contain both reference models which are unique for one application area (e.g. different car parts) and topical models which define entities that are shared by a number of application areas, for instance distribution networks, finite element analysis and presentation draughting.

The AEC-committees of PDES and STEP have up to date proposed the following application models:[9]

- Building Systems Model
- Ship Structures Model
- Ship Outfitting Model
- Plant Design Model
- Distribution Systems Model
- General AEC Reference Model

The first draft proposal for STEP version 1.0 was adopted in December 1988[10]. At present it is expected that STEP might reach the status of an international standard by 1991 or 1992.

5. 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. [11] 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 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 and documentation practice.

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 data model used in the RATAS-product model follows the entity-relationship data model quite closely.

So far, agreements have only been reached on the general priciples of the product model (use of object-centered concepts, main abstraction hierachy et.c). The detailed definition of object classes, attributes and relationship types remains to be done.

The RATAS model defines an abstraction hierarchy [12] taylored to the needs of building designers. The five abstraction levels are: building, system, subsystem, part and detail.

The **building object** contains attribute data about the site, the climate, the total size of the building, the construction cost, the type of building et.c.

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, et.c.).

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

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

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 quantites and specifications. The most important attribute value types in the RATAS-model are numeric values, text and 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.

The **part-of relationship** specifies that a particular object belongs to a larger object. No other information is

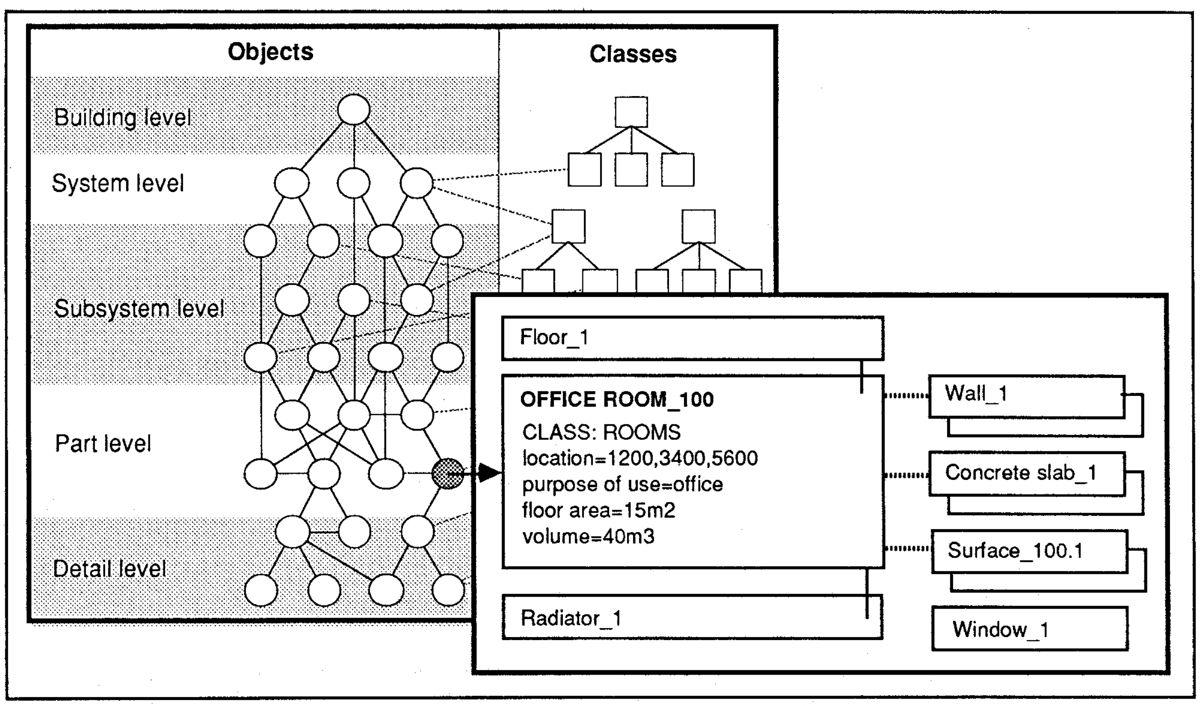

Figure 1: Overall structure of the RATAS Building Product Model.

connected to the part of relationship. Mostly this type of relationship connects objects from different abstraction levels.

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 relationships can in fact be treated as objects of their own.

At present there are several research and development projects going on in Finland in which attemps are made to develop further the principles defined during the RATAS-project. In particular the Technical Research Centre of Finland has built prototype building product models on a modest scale using such tools as relational databases [13] and hypermedia [14].

In the near future a number of application programs will be built, which will be integrated with the product model itself. The applications deal with checking the compliace of a design with the Finnish fire regulations, the taking off of quantities for construction cost estimation and with calculating the energy consumption of a proposed building design. All of these application programs will internally use conceptual data structures compatible with the main product model. Looking at the example of a particular wall it is evident that cost estimation, fire protection checking and energy balance calculation have some data needs in common (dimensions of the wall, information about the material) but differ in other information needs (price of material, fire rating, U-value). It is the common information needs which need to be dealt with in the building product model itself.

6. Object-oriented Energy Simulation Programs

The product model approach outlined above is at present being developed by specialists interested in the exchange of CAD/CAM data, and concerns all fields of industry. Quite independently people involved in the development of methods and software for building energy simulation have taken an interest in the possibilities offered by knowledge-based and object-oriented software techniques. Several prototype systems and proposals have in the last few years been presented.

The basic aims of these can be summarized as follows:

- To make the development and validation of energy simulation programs simpler and more efficient through standardization and through the definition of reusable program elements which can be included in different application programs.
- To make the use of such simulation programs easier for non-experts through a better quality of the software, better user interfaces, more transparent calculation methods et.c.
- To facilitate the integration of energy simulation programs with CAD software and CAD databases, thus lowering the threshold for using simulations in everyday design work.

In prototype work for energy simulation knowledge-based and object-oriented software tools seem very useful compared to traditional algoritmic programming methods. One of the major advantages of using the frame and object data models is that the conceptual data structures are highlighted very clearly, which makes it easier to develop, document and maintain software. Another very strong feature is the inheritance of properties between classes of objects, due to which a lot of computer code can be reused with minor revisions for different types of components et.c.[15]

A fundamental dilemma which has to be solved if the integration with CAD-software is to be achived is the automatic generation of the concepts used in energy simulation programs (parameters and equations describing the thermal characteristics and behaviour of the building and its components) from the concepts used by architectural designers (spaces and physical building components, and their material properties). A convenient way of handling this is the technique called data abstraction, a technique which was described earlier in connection with the RATAS-project. Data abstraction implies that it is possible to wiew the same data base in different ways, depending on the particular needs of the information's user. It also implies that lower level basic entities can be collected into higher level system components in flexible ways. The same basic entity may for instance belong to several different system level components. This is not easy to achieve with conventional data files or hierarchical database systems, but is a typical feature of the more complicated data models presented earlier.

The most well-known example of a general standardized library of reusable software modules for energy simulation is the proposed Energy Kernel System.[16] EKS is in fact primarily concerned with the standardization of calculation algoritms and methods, but it also touches on issues related to the more static product modeling of a building.

Examples of systems currently under development which try to reconcile the component wiew of buildings with equation set modelling are SPANK [17] and MODSIM [18]. The integration of energy simulation software with current commercial CAD systems [19] and with a geometric modeller based on the product model approach [20] have been studied in a couple of projects.

7. Conclusions

In conclusion there seems to be a need for more cooperation between developers of state-of-the-art building design CAD-systems and developers of energy simulatation methods and software. Developers of building product models would need to focus more on the distinction between physical building entitities and abstract (equation) entities and how to treat these in their product models. Developers of energy simulation programs and especially developers of possible standards for such software should try to incorporate relevant entities defined in general product models into their own knowlegebased and object-oriented systems. They could also benefit from a clear understanding of the difference between the conceptual analysis of data structures from their implementations using different software tools, and from using the formal EXPRESS language used in the development of STEP as a common software-independent data definition format.

References:

1 Penttilä, Hannu; Björk, Bo-Christer
Data Structures in Computer-Aided Building Design
Building Book Ltd, Helsinki 1989, 75 p.

2 Smith, Bradford
International Efforts in Product Data Exchange
Proceedings of the MICAD86 conference, Hermes, Paris, 1986, pp. 745-751

3 Björk, Bo-Christer
Basic Structure of a Proposed Building Product Model
Computer-aided design, Vol 21/2, march 1989, pp. 71-78

4 Chen P.
The entity-relationship model: Towards a unified view of data, ACM Transactions on data base systems Vol 1/nr. 1, 1976, pp. 9-36

5 Stefic, Mark; Bobrow, Daniel G.
Object-oriented programming: Themes and variations
AI Magazine, Vol 6/no. 2, p. 40-62

6 Aish, Robert
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, 24-26.10.1988, 5p.

7 Henson, Troy; ed.
Artificial Intelligence and Simulation: The Diversity of Applications
Simulation Councils, Inc.,San Diego 1988, 275 pp.

8 Schenck, Douglas
Express, ISO TC184/SC4/WG1, Draft N 210, Febr 1988, 77 p.

9 Gielingh, Wim
Computer Integrated Construction: a major STEP forward
Proceedings of the ARECDAO89 conference, ITEC, Barcelona 1989, pp. 29-48

10 Wilson, P.R.; Kennicott, P.R. eds.
Integrated Product Information Model (STEP 1.0)
ISO TC184/SC4/WG1 Doc nr. N 284, 27.10.1988, 640 p.

11 Enkovaara, Esko; Salmi, Markku; Sarja, Asko
RATAS Project - Computer Aided Design For Construction
Building Book Ltd, Helsinki 1988, 62 p.

12 Eastman, Charles 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, pp. 335-337

13 Björk, Bo-Christer; Penttilä, Hannu; Saarinen, Heikki; Moisio, Jorma; Finne, Christer; Nervola; Maija
A Prototype Building Product Model Using A Relational Database, ARECDAO89 Conference, ITEC, Barcelona 13-14.4.1989, pp. 101-117

14 Penttilä, Hannu; Björk, Bo-Christer
A Scenario for the Development and Implementation of a Building Product Model Standard, Paper presented at the symposium on Current Research and Development in x Integra
ted Design, Construction and Facility Management, CIFE, Stanford University, California, 28-29.3.1989, 17 p.

15 Liekens, Johan
Energy Simulations Using an Object Oriented Approach
Draft paper, Vrije Universitet Brussel, 20.2.1989, 8 p.

16 Clarke, J. A.; Irving, A. D.; Lockley S; Mac Randal, D
An Object-Orientated Approach to Building Performance Modelling, User 1, A working conference for users of simulation hardware, software and intelliware, Society for Computer Simulation, 6-8.9.1988, Oostende, Belgium, pp. 144-151

17 Sowell, E. F.; Buhl, W. F.; Erdem, A. E. ; Winkelmann, F.C.
A Prototype Object-Based System for HVAC Simulation
Proc. System Simulation Conference, University of Liege, 1986, p. 471-491,

18 Sahlin, Per
MODSIM - a Program for Dynamical Modelling and Simulation of Continuous Systems
Unpublished paper, Swedish Institute of Applied Mathematics, Stockholm, 1988, 28 p.

19 Haakana, M.; Heikkinen, L.; Kallavuo, M.
Energy Calculation Program for Early Stages Utilizing Geometric Data from CAD systems
CIB 5th International Symposium on the use of computers, Bath England, 7-9.7.1986, Gibse, England, pp. 68-71

20 Gielingh, Wim
Analysis of Energy-Consumption and Radiator-Selection for Rooms in a Building
Unpublished paper, TNO-IBBC, Delft, Netherlands, february 1989, 4 p.