Building and Environment, Vol. 27, No. 2, pp. 173–194, 1992.
Printed in Great Britain.

0360–1323/92 $5.00+0.00
© 1992 Pergamon Press Ltd.

A Unified Approach for Modelling Construction Information

BO-CHRISTER BJÖRK*

Computer Integrated Construction (CIC) is currently a priority subject area for information technology research in the construction industry. Published research results have so far tended to be either verbal discussions of the overall concept of CIC and its potential benefits, or descriptions of some restricted aspects of the overall CIC process using formalized data or activity modelling tools.

This paper attempts to extend the scope of such formalized approaches by suggesting how object-oriented conceptual modelling techniques, which until now have been used mainly for product modelling, could be used to define a framework for the overall information management in a construction project. Such a framework could be used as a basis for the development of more detailed standards, placing them in a more general context. It could also help in setting strategies for the development of construction computing applications, which in due time will become parts of a CIC environment.

1. INTRODUCTION

THE concept of computer-integrated construction (CIC) emerged during the latter half of the 1980s, no doubt influenced by the CIM-efforts of other industries, but also based on a fifteen-year-long research tradition in computer-aided building design [1]. Computer-integrated construction, in close conjunction with on-site automation and the use of industrial construction systems based on extensive prefabrication, has been proposed as one of the means the construction industry has of increasing its productivity and solving problems related to the quality of its products [2].

A number of authors have tried to outline the definition of CIC in overview articles discussing its components and benefits [3, 4]. The definition used as a basis for this paper includes both information management and construction activities. Thus computer-integrated construction consists of both the use of information technology in the different phases and tasks in construction, and the integration of these phases and tasks through the use of digitally stored data and data transfer.

Among the elements needed to create a computer-integrated construction environment are:

- Advanced personal workstations and peripheral hardware;
- Application software;
- The extensive use of data bases for storing design and project management data;
- Intelligent user interfaces;
- An infrastructure of data transfer networks;
- Data structuring and transfer standards;
- Data and knowledge bases with general construction information.

A number of research and development projects have dealt with limited aspects of CIC using formalized modelling techniques. In some of these projects the construction process has been modelled using activity models. In other projects parts of the overall information used during a construction project has been modelled using data flow or conceptual data modelling techniques. Examples of such formalized models are provided by GARM [5], IBPM [6], BIM [7], ISOBAT [8]. The aim of this paper is above all to suggest how such formalized techniques could be used more extensively to cover the totality of the information produced during the design, construction and maintenance of a building. The list of the elements of CIC presented above could thus also include what one might call framework or generic construction process information models.

- Framework models to support CIC development and standardization.

Using object-oriented conceptual models to structure information concerning both the building to be designed, the construction project's participants, robots, construction activities, cost and contract information, technical calculation results, design documents etc. could offer significant benefits. The most important one is the ability to formally describe the relationship between the data contained in the information systems of different types of participants in AEC projects. By extending data transfer and retrieval capabilities from coherent types of information (currently the emphasis is on CAD-data and EDI messages), to a more comprehensive coverage of heterogeneous construction information, these relationships offer the means to obtain the flexible information retrieval and management needed in CIC.

* Laboratory of Urban Planning and Building Design, Technical Research Centre of Finland, Itätuulenkuja 11, 02100 Espoo, Finland.

Since the overall framework proposed in this paper is conceptual, it becomes possible to extract submodels, in the form of database views, relating to the information needs of particular agents in the process. Thus the overall model should contain the semantic content of many current models (among others, traditional building classification systems [9] and emerging building product models) as subsets, but should also offer a tool for the description of the integrating information between these—a crucial element in computer integrated construction.

2. SCOPE OF THE FRAMEWORK MODEL

Information used in the construction process can, from different viewpoints, be divided into a number of categories.

The first viewpoint can equally well be applied to any information system and isn't specific to construction. Information can first of all state facts. This category of information is what design documents, which are the results of design decisions, primarily concern. Information that needs to be transferred between computing systems in the construction process is mostly of this type. Examples are:

- The color of the surface x of wall y is green;
- The price of a door of type xz is 2050 marks.

Secondly, information can define goals and requirements which a particular project must fulfil. Typical assertions of this type are:

- the total construction cost may not exceed 120 mill. marks;
- the building may not have more than 5 floors;
- the U-value of the outer walls must be less than 0.25 W/k m^2.

This kind of information could be called requirements or constraints. The term 'requirement' seems more natural when discussing construction practice, whereas constraint has a clear meaning in database theory. Requirements can, for instance, be derived from building regulations or the client's goals. Quality assurance systems are very much concerned with this type of information. Constraints are typically defined using operators which restrict attribute values of building description objects within certain limits or to certain predefined values.

The third category of information states rules which restrict facts, but which apply in general and are not tied to a particular project. Two examples are given below:

- If the building has no more than 2 storeys, or is less than 7 metres high and has an area of less than 400 m^2, it can be constructed to be fire retarding;
- a beam, which is directly or indirectly structurally supported by a column, cannot be erected before the column.

These three categories of information could be called 'facts', 'constraints' and 'knowledge'. From a programming language viewpoint, facts can be constructed using simple assignment statements, requirements are represented predominantly by inequality operators and knowledge can be formalized by the rule formalisms of knowledge-based systems. This analysis is quite simplistic, but should be adequate for stating the scope of our modelling effort. The choice of scope influences the choice of conceptual modelling tools to be used.

The second semantic viewpoint divides information into project-specific and more general information. The fact that 'room 103' with a given location and floor area is included in the design of a particular building is clearly project-specific. The description of the structure of the walls enclosing 'room 103' may on the other hand be given as a reference to a database with standard details, either defined within a particular firm or available from a construction information service.

Facts can be both project specific and general. Constraints are predominantly project-specific. Knowledge is usually general in nature.

The third viewpoint concerns presentation and categorizes the types of documents, which are used to present the information for human interpretation. Some typical presentation formats used in construction are:

- Drawings;
- Schematic drawings;
- Realistic visualizations;
- Written specifications;
- Calculation results, for instance in tabular form;
- Lists of components, bills of materials;
- Contracts, orders and bills.

Using current software technology, different categories of software are best suited to handle each type of presentation. Drawings and schematic drawings are handled by computer-aided design and drafting systems. For some schematic drawings (i.e. timetable diagrams) customized software is better suited. Realistic visualizations using raytracing, paint techniques, etc. often use CAD-data as input, but may need special types of software. Specifications are mostly produced using word processing software. Calculations are carried out using spreadsheets or customized software written in scientific programming languages. Orders and bills can be handled by word processing software, but are often generated and processed by database sysems.

Using the categorization of construction information presented above, we limit outselves in the rest of this paper to the modelling of project specific facts and focus on the semantics of the information—thus covering all different presentation types.

The information which needs to be communicated to other parties in the construction process consists mostly of factual information and not of the constraints and knowledge used in the process of determining this factual information. Clearly constraints are very important in the early briefing stages of projects and in quality assurance applications. Research is needed to study their modelling through conceptual modelling techniques. The modelling of construction knowledge is a vast research area of its own (see for instance Kalay [10], Coyne *et al.* [11]). Knowledge will mainly reside in application programs and expert systems, and will have less effect on the actual transfer of data between project participants. Requirements and knowledge have consequently been deliberately omitted from the scope of the model presented in this paper.

The focus on project-specific rather than general information is due to our primary concern to provide a framework for information management within construction projects. It would be a fairly straightforward task to extend the analysis to general construction databases containing information about materials, average costs, building regulations, etc. This would, however, necessitate the inclusion of knowledge in addition to facts in the analysis.

Since the focus in the conceptual framework is on the semantics of the information, we abstract away from the presentation format of the information. The same object-oriented description of a wall can be used as a basis for presenting information about the wall in a CAD-drawing, standard detail or spreadsheet table, as exemplified in Fig. 1.

3. CURRENT MODELLING APPROACHES

There seem to be at least three major approaches for the use of formal modelling techniques in the description of the information management and physical processes in a construction project. The first approach focuses on the modelling of activities in a project and their causal relationships using graphical diagrams. Such diagrams are easy to explain to people from industry, and can help in improving construction management procedures. They can also provide a basis for defining computer applications. In this respect such diagrams are close in spirit to the traditional algorithmic way of defining computer systems. The shortcomings of this approach for our purpose will be discussed later on this paper.

In the second approach, the data flows between different information processing activities are identified and shown in data flow diagrams. This approach is often used in projects with industry participation. In the short term this approach could provide good results, but in the long run it tends to encode the conventions of today's industry practices into the applications, thus becoming something of an obstacle for developing the process and the data management methods. The number of data flows will also quickly become quite large, especially in project phases which involve a lot of two-way communication between project participants.

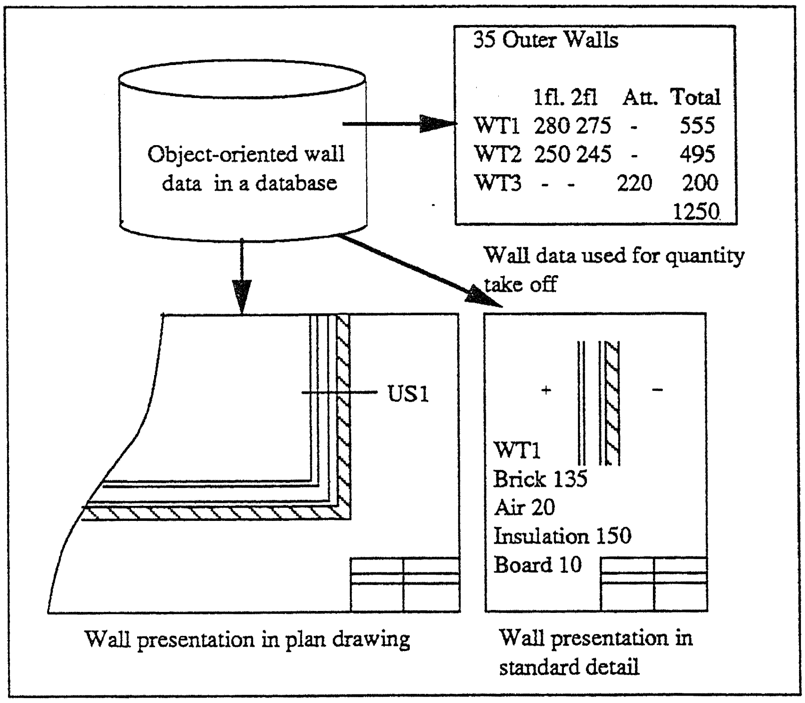
The third approach focuses on modelling the structure of the information describing the products, processes, resources and other elements of the construction process using conceptual modelling techniques. This approach has been particularly popular in product model research, but is also starting to have an impact on knowledge-based project management research. This approach aims at determining the data structures used for storing construction data in databases.

In Section 5 some examples of the use of these different approaches will be given. In order to analyze these examples in a unified way, and for the presentation of the overall framework, concepts from database theory are needed. These are explained in the next section.

4. DATA MODELS, CONCEPTUAL MODELS

4.1 *Fundamental concepts*

In the following, some fundamental concepts from database theory are presented. For a more detailed



35 Outer Walls

	1fl.	2fl	Att.	Total
WT1	280	275	-	555
WT2	250	245	-	495
WT3	-	-	220	200
				1250

Object-oriented wall data in a database
Wall data used for quantity take off
US1
+
-
WT1
Brick 135
Air 20
Insulation 150
Board 10
Wall presentation in plan drawing
Wall presentation in standard detail

Fig. 1. The object-oriented structuring of data supports many different alternative presentation formats and uses of the same information.

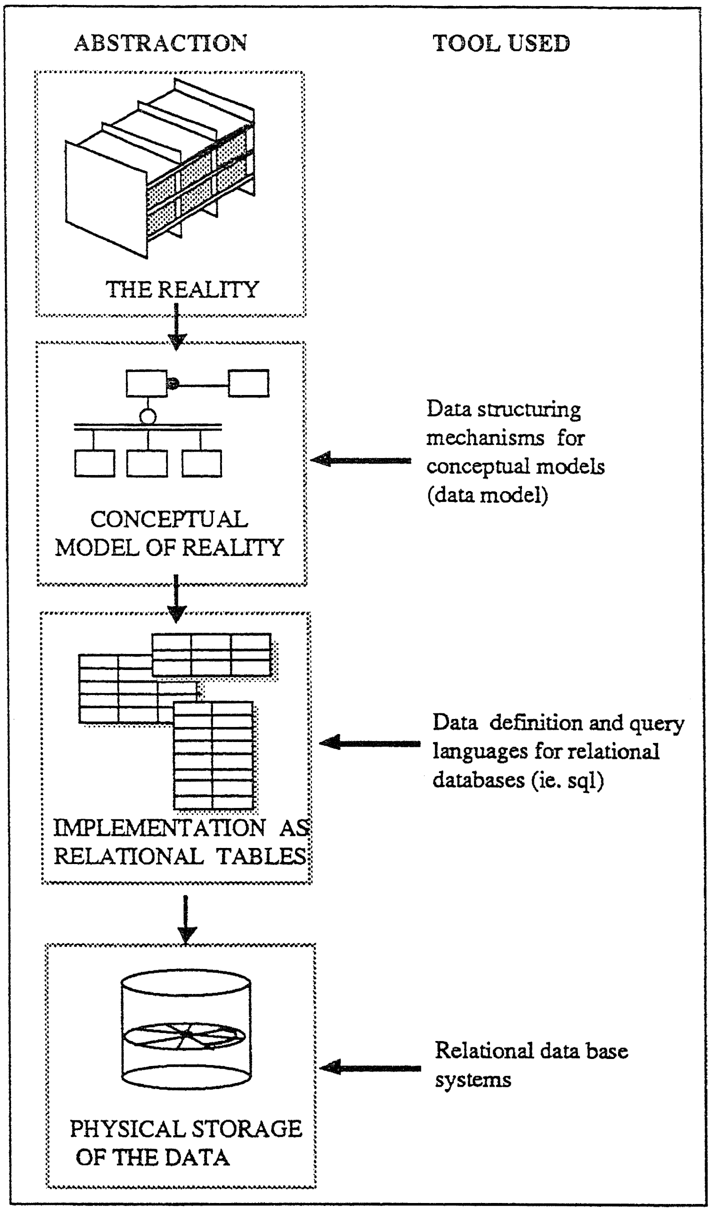

ABSTRACTION
TOOL USED
THE REALITY
CONCEPTUAL MODEL OF REALITY
Data structuring mechanisms for conceptual models (data model)
IMPLEMENTATION AS RELATIONAL TABLES
Data definition and query languages for relational databases (ie. sql)
PHYSICAL STORAGE OF THE DATA
Relational data base systems

Fig. 2. A fundamental idea in data base theory is to clearly separate the conceptual modelling of the concepts and relationships needed to describe some aspect of reality from the representation formats used for storing and manipulating this information.

description, any good textbook on database theory can be consulted. More theoretical presentations discussing the similarities and differences between different techniques can be found in Brodie [12], Peckham [13] and Hull [14].

A data model provides the basic data structuring mechanisms for describing the data, relationships and constraints of the information stored in any information system. An analogy would be the basic grammar utilized in natural language.

Using a specific data model, conceptual models can be built. A conceptual model specifies the categories of information used in a specific domain or database. In a conceptual model only the information itself is modelled, not the format in which the information is stored or presented. This in turn could be specified in physical data models, which can be on different abstraction levels. On the most basic physical level, information storage as binary digits in the specific locations in the computer's memory has to be specified. In the early days of computing this had to be encoded in application programs, but today operating systems and database management systems take care of this. Figure 2 tries to illustrate the relationships between these concepts in the case of relational data bases.

It should be noted, that the use of the term data model is quite inconsistent in the literature describing the use of conceptual modelling techniques. Many authors use the term 'data model' as a synonym for 'conceptual' model. In this paper a clear distinction is made between data model and conceptual model.

The basic concept used in almost all data models is the object or entity. An object is a set of interrelated data about some 'thing' in the modelling domain. The data structures used in particular objects are defined in classes, which provide templates for inserting the data values of particular objects. Often different classes can be interrelated via inheritance structures. Similar concepts to objects are frames in knowledge-based systems, abstract data types in programming languages and the 'objects' of object-oriented programming languages.

Objects can be interrelated via relationships. The object describing a door in a building may, for instance, have relationships to the objects representing both of the rooms the door leads to, and also to the object representing the wall into which the door is fitted. Relationships may be further classified according to their semantic content (e.g. pipe connection), or according to the consequences of deleting objects from database (existency dependency).

Some, but not all data models also contain the concept of 'attributes'. An attribute can be used to specify a property related to an object (a door's height for example). In some data models attributes can only be primitive data types (integer, real numbers, alphanumeric strings), but other data models allow objects or links to them to function as attributes of other objects.

In the practical definition of conceptual models, it is often very difficult to decide whether to model a specific data item as an object, attribute or relationship.

Additional data structures which can be included in data models are rules, integrity constraints, methods, etc. Recently there has been a growing interest in object-oriented databases [15] which utilize a quite comprehensive set of data structuring mechanisms.

4.2 *Abstraction principles*

The basic idea behind abstraction is to highlight and hide different types of information about things which have been modelled, depending on the level of the view. Abstraction can work using aggregation, through a successive decomposition of a product into smaller parts. It can also work, using the mechanisms of generalization and specialization, for focusing on different data describing the same real world objects.

In the higher abstraction levels of a decomposition hierarchy we find information about aggregate objects (surface area of the total building, duration of the total construction process etc.), on more detailed levels we get information about the sub-objects that comprise these aggregate objects, and which often implicitly determine the attributes of the aggregate objects. An attribute of the building object might, for instance, be the number of floors. This can actually be calculated by checking the number of floor objects on a lower abstraction level.

Applied to conceptual models of construction information, the abstraction principle can be used for the definition of abstraction hierarchies for different kinds of information:

- The building from its detailed components to the whole building (this is commonly called a building product model).
- Project activities from detailed microscopic activities to the whole project.
- Project cost from detailed cost items to overall construction cost.
- The project's organization from individual participating firms, their personnel and machinery to the overall project organization.
- Project documentation from simple detail drawings, individual bills and meeting protocols to overall project documentation.

In addition to decomposition, abstraction can also be used for generalization-specialization. This is usually accomplished through some kind of subtyping or inheritance mechanisms between object classes. Different species of animals could, for instance, be modelled using this mechanisms. The basic idea is to specify all information on as general a class level as possible and then to define additional distinguishing parameters on the subclass level.

4.3 *Conceptual modelling tools*

Related to data models are explicit or implicit data modelling tools using the data model in question. The two main types are graphical schema languages (for instance entity-relationship, IDEF1X or NIAM diagrams) and data definition languages. The former provide human users with a good overview of data relationships in much the same way as plan or schematic drawings are very useful visualization aids to construction professionals. Data definition languages are integral parts of data base systems, AI-tools and programming language syntaxes and are directly interpreted by software for the creation of the necessary physical storage mechanisms.

The modelling tool used in this presentation is intentionally one of the graphical tools used in the international standardization effort on product data models; STEP [16]. Operationally the data model in which STEP is defined has been implemented in the EXPRESS data definition language [17], which in itself is a part of the STEP-Standard. Either one of the graphical data modelling languages IDEF1X [18] and NIAM [19] can be used in the presentation of STEP-submodels.

The choice of data model to be used in a conceptual modelling task is dependent on the type of software used for implementing applications based on the model. It is also, to some extent, a matter of taste. Many researchers involved in STEP-work prefer NIAM to IDEF1X. This is partly due to the fact that NIAM doesn't distinguish between entities and attributes in the same sense as IDEF1X does which can sometimes be an advantage, especially in early phases of conceptual modelling work. On the other hand, NIAM easily results in complicated, weblike schemas. An example of a NIAM schema, describing the components of a water radiator network, is given in Fig. 3 [20]. IDEF1X, on the other hand, is particularly well suited if an implementation is to be done using relational database techniques.

The most important thing in the development of a framework model of the type described in this paper is to choose one data model, with sufficient semantic

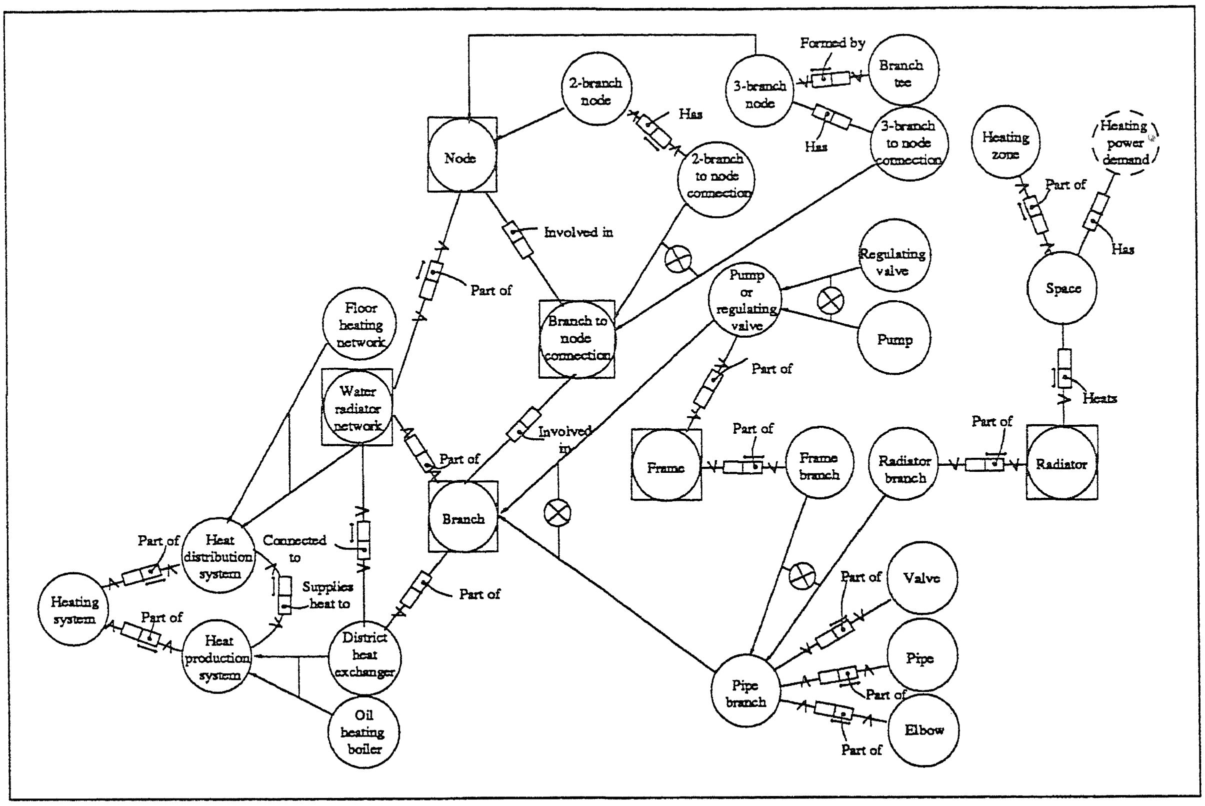


Fig. 3. NIAM schemas tend to become quite complicated as this conceptual model of a water radiator network shows.

functionality, and apply it consistently to the modelling of all kinds of information. The same observation has also been made by Böhms in his work on a CIM reference architecture [21]. In Böhms' model, which has been tested with case studies concerning the manufacture of ship propellers and elevators, a graphical data modelling method based on NIAM (the 'CIM Base Model') has been used. In addition to the generalization/specialization abstraction included in NIAM, Böhms has added a specific graphical notation for the aggregation/decomposition abstraction. In standard NIAM models the same could be achieved using relationship labels such as 'part of'. Böhms CIM Base Model also contains a handful of generic entity types.

If a conceptual model is formulated in one particular data model, it is possible to translate at least part of the data structures to other data models. In STEP work this constantly needs to be done in the translation of NIAM and IDEF1X schemas to EXPRESS definitions. Such a conversion task is straightforward if the latter data model contains the data structuring mechanisms of the former data model as a subset. If this is not the case, some semantics may be lost.

In this presentation IDEF1X has been used. The basic concepts used in IDEF1X are shown in Fig. 4. The data model underlying the graphical notation of IDEF1X is the entity-relationship model [22] with some data structures typical of relational databases added.

The basic elements in IDEF1X are entities, relationships between entities and attributes. In IDEF1X schemas, entity types are described by rectangles, relationship types by lines connecting rectangles and attributes by named fields within the rectangles. It is very simple to define attribute data in IDEF1X. Attributes can also function as keys, which identify separate entities from each other.

IDEF1X offers several different data structuring mechanisms. In the schemas presented in this paper, the important mechanisms are the generalization/specialization abstraction (subtype/supertype) and the ability to indicate the cardinality (through the use of dots) and type (use of labels on arrows) of relationships. 'Cardinality' means the number of entities participating in a particular relationship. The main types are one-to-many and many-to-many. IDEF1X has some restrictions on the use of cardinalities, which are not discussed in this presentation.

One of the advantages of the IDEF1X methodology is the practice of naming entity types with nouns and relationships with verbs. This facilitates reading IDEF1X schemas.

5. EXAMPLES OF THE USE OF DIFFERENT MODELLING APPROACHES

5.1 *Activity diagrams*

Let us consider a typical activity diagram of construction activities. SADT [23] and the closely related IDEF0 are currently widely used graphical techniques

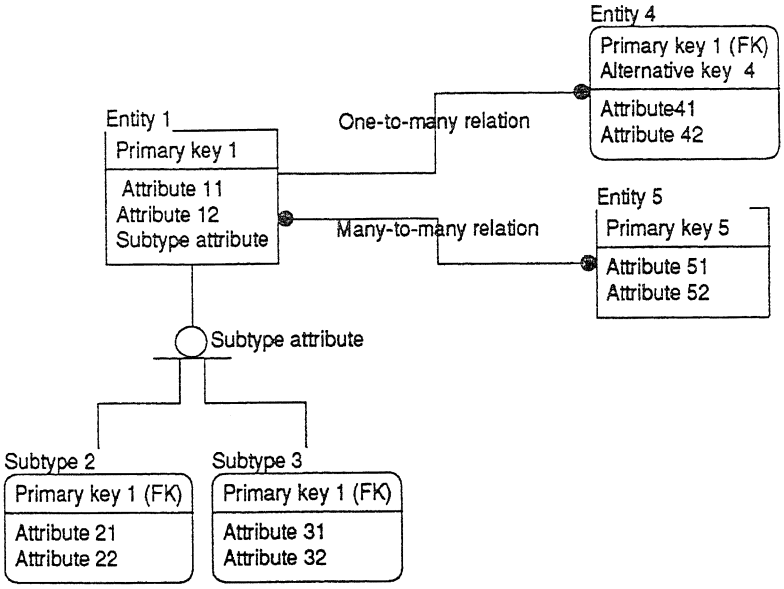


Fig. 4. In IDEF1X schemas entity types are described by rectangles, relationship types by lines connecting rectangles and attributes by named fields within the rectangles. Subtyping is also supported.

for activity modelling. The use of such diagrams has a clear cognitive function, since it is easier for us to understand the causal relationships between different activities if they are modelled using graphical schemas. In such schemas the positioning of boxes in relation to each other and arrow symbols are used to convey information about the logical relationships between activities. An example of a typical SADT-diagram is shown in Fig. 5.

What we actually see depicted in SADT or IDEF0-diagrams is, contrary to IDEF1X and NIAM diagrams, not a conceptual schema of an underlying data structure for the description of activities, but an instantiation with real data about a particular or typical process. An activity

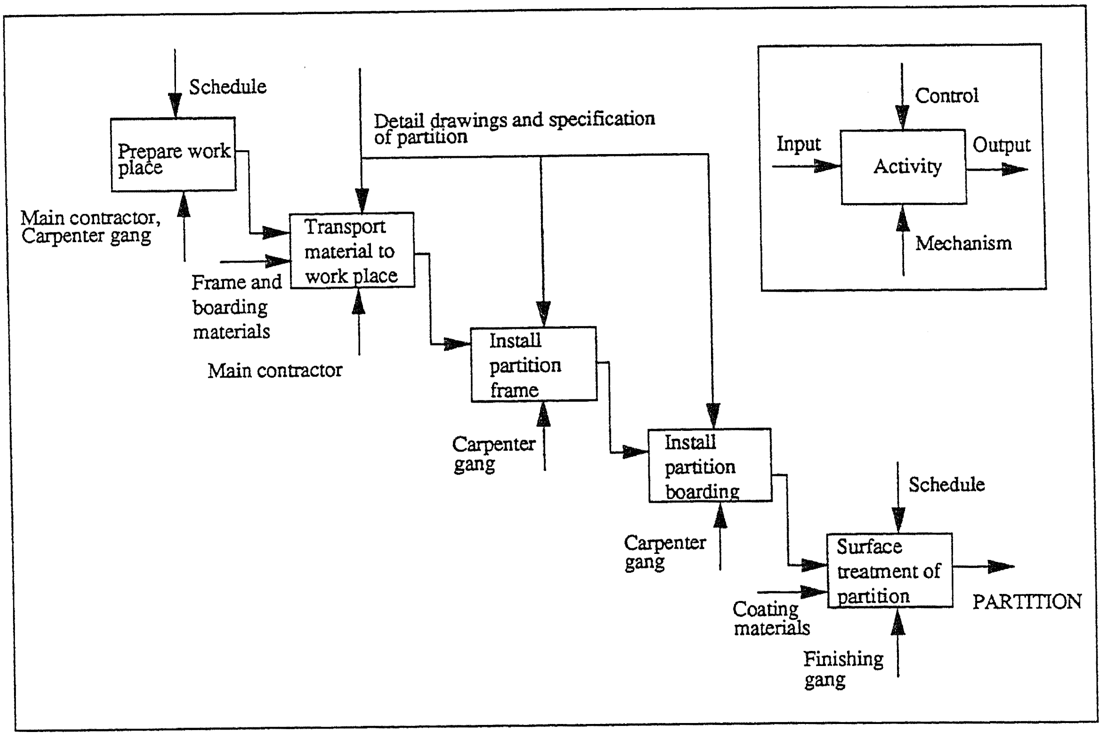

Schedule
Prepare work place
Main contractor, Carpenter gang
Detail drawings and specification of partition
Transport material to work place
Frame and boarding materials
Main contractor
Install partition frame
Carpenter gang
Install partition boarding
Carpenter gang
Schedule
Surface treatment of partition
Coating materials
Finishing gang
PARTITION
Control
Input
Activity
Output
Mechanism

Fig. 5. In SADT- or IDEF0-diagrams, the positioning of boxes in relation to each other and arrow symbols are used to convey information about the logical relationships between activities.

diagram could thus not be used in a straightforward fashion to implement database systems storing information about the activities of a process.

In a first analysis, the implicit conceptual model of SADT seems to consist of the following object classes: activity, input, output, control, mechanism. The function or activity is the central object class and a specific graphic symbol (rectangle) has been reserved for it in the SADT diagrams. Objects of the other classes are only depicted by labels associated with the arrows in the diagram. These labels can be explained in a glossary.

In a closer analysis, it might be more fruitful to consider these arrows as symbolic representations of relationships between object classes which are only implicit in the diagrams. Thus input, output, control and mechanism could be interpreted as relationships between an activity and some very generic class representing agents, information, products etc. A suggestion for the conceptual schema (in IDEF1X-notation) underlying an SADT-diagram is shown in Fig. 6. In the diagram the generic class representing anything that can function as either output, input or control has been called 'OIC' (output and/or input and/or control). The reader should be cautioned that this is just one possible interpretation.

Böhms [21] used the results formalized in IDEF0 of his analysis of two CIM-process case studies as a basis for the formulation of his application-specific CIM reference architectures using the NIAM-based 'CIM base model'. Although not stated explicitly in his work, this implicitly necessitates the formulation of the conceptual model of the IDEF0 diagrams in the CIM base model.

There is no support for inheritance between object classes in SADT as this is actually not needed, since no attributes are defined. Also the number of object classes is so small that there is no need to subtype these in any way. The modelling methodology does, however, support decomposition abstraction through a strictly hierarchical organization of models and submodels. Each activity may be subdivided into a restricted number of subactivities which are shown in separate subdiagrams. Inputs and outputs may also be shown in greater detail in subdiagrams.

Consequently, it can be seen that the conceptual model of SADT is rather limited. Although results of activities can be modelled in their function as inputs or outputs, there is no possibility to include more information about these or to model their relationships to other object classes. (It should be noted that SADT also contains other diagrammatic methods for data modelling which, however, are used more seldom and are not analyzed here.)

5.2 *Data flow modelling*

Data flow modelling techniques have been used in a number of projects for analyzing the information flowing between different information processing activities in the design and construction process. A popular type of data flow diagram used for this purpose is the Structured Analysis Method [24]. In these diagrams activities are represented by circles and data flows by arrows between the circles. It is also possible to use a symbol for temporary data stores.

This technique has, for instance, been used for the systems analysis preceding the development of a relational database system for the management of the design and manufacturing process, EMTRIS [25]. It has also been used in a recent Danish study of the information flows in the construction bidding phase, DigiDok [26]. An

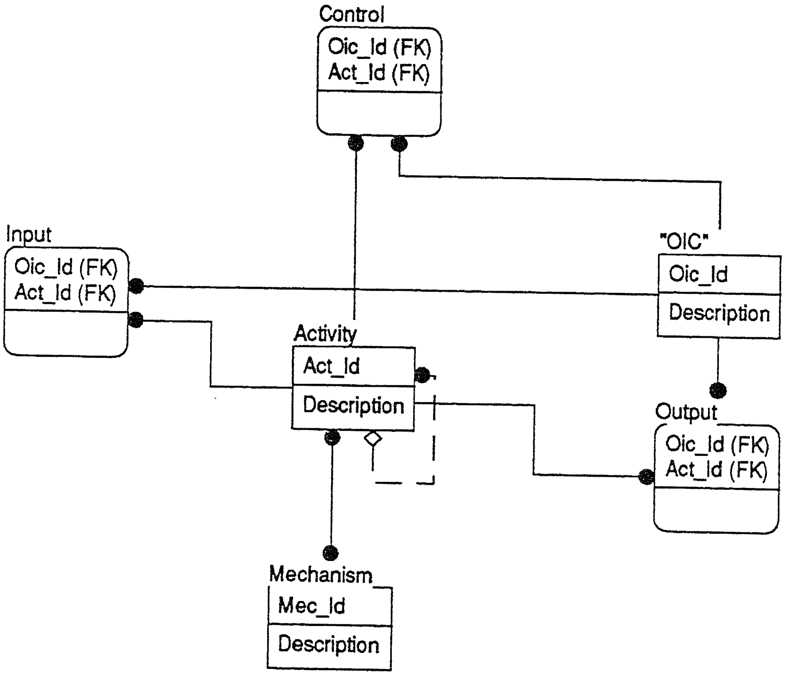


Fig. 6. One possible interpretation of a conceptual model underlying SADT diagrams is shown in this IDEF1X schema.

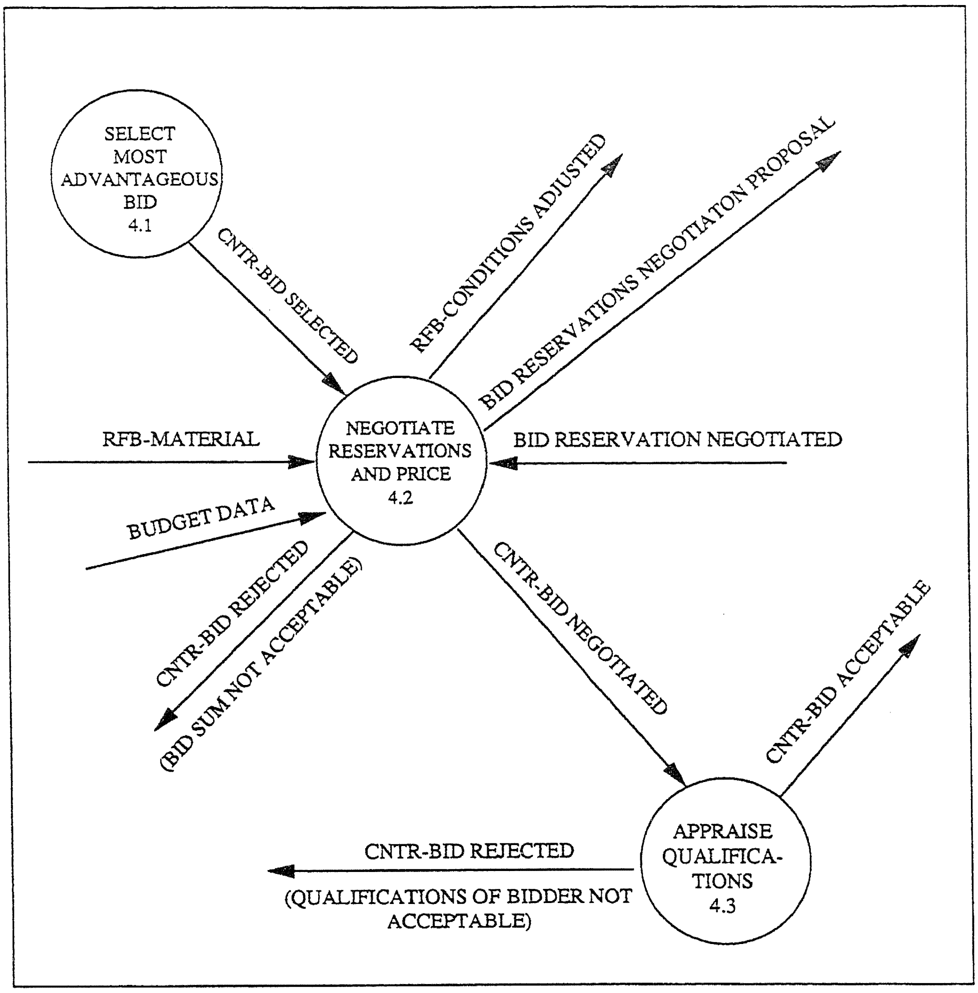

SELECT MOST ADVANTAGEOUS BID 4.1
CNTR-BID SELECTED
RFB-CONDITIONS ADJUSTED
BID RESERVATIONS NEGOTIATON PROPOSAL
RFB-MATERIAL
NEGOTIATE RESERVATIONS AND PRICE 4.2
BID RESERVATION NEGOTIATED
BUDGET DATA
CNTR-BID REJECTED
(BID SUM NOT ACCEPTABLE)
CNTR-BID NEGOTIATED
CNTR-BID ACCEPTABLE
APPRAISE QUALIFICA-TIONS 4.3
CNTR-BID REJECTED
(QUALIFICATIONS OF BIDDER NOT ACCEPTABLE)

Fig. 7. An example taken from the Danish DigiDok project of the use of data flow diagrams for modelling information management in construction.

example of the diagrams produced in the Danish project is given in Fig. 7.

The conceptual model underlying the Structured Analysis Method is simpler than the implicit conceptual model of activity diagrams. It only contains two object classes : processes and data flows. Temporary data stores and system terminators could be added, but are less important. It would therefore be relatively simple to convert structured analysis diagrams to an object-oriented representation containing exactly the same information.

5.3 *Modelling of the contents of documents*

A further category of modelling deals with the standardization of the content and syntax of specific types of documents transmitted digitally in the construction industry. So far this type of standardization has mainly been concerned with trade documents, which are relatively simple in data structure, and of which great numbers are transmitted daily. In Europe this work is organized as EDIFACT work, covering all branches of industry.

A special syntax has been defined for EDI-messages. In addition, the specific data contents of different message types (purchase order, bill, etc.) have to be defined. The messages themselves are not intended for direct human interpretation, but are transmitted between pre- and postprocessing software integrated into the accounting, procurement and other such applications of different firms.

Standardized EDI-messages provide an example of a successful use of the data flow modelling approach on a micro level. The success is, in this author's opinion, largely due to the fact that EDI formalizes relatively simple messages or documents, which already in their paper version are highly standardized. The transmittal of full building designs as standardized EDI-messages would be a totally different story. People working within the EDI-community have realized this. Thus a suggestion for the EDI transfer of construction drawings would only include general information about drawings in the EDI-message. The actual graphics could then be transferred using special purpose standards such as DXF, or future STEP application protocols for 2-D drawing exchange [27].

Another interesting development is the use of the ISO Standard Generalized Mark Up Language (SGML) [28]. SGML was originally developed to identify the inherent structure of documents by inserting invisible but retrievable markers in the text; for instance for sections, abstracts, authors etc. Often these are indicated in text by different fount types and sizes, underlining, indentation, etc., but SGML separates the semantic definition from the syntactic presentation (compare with the idea of conceptual model versus physical data model from database theory). For highly structured document types, special document type definitions (DTD) can be created.

In addition to the automatic generation of typesetting instructions another interesting use of SGML is described by McKell [29]. By defining the text strings between specific markers as entities in a conceptual model, it becomes possible to mark written documents automatically or ex-post and create knowledge-base information from them.

The use of SGML for the structuring of hypermedia applications of building regulations is also currently being studied [30]. Compared to a full-fledged expert system approach, this would in the short run provide a more feasible approach for the development of efficient data retrieval systems for regulations users. An example of the use of SGML for this type of application is given in Fig. 8.

Both EDIFACT standard messages and SGML document type definitions thus contain implicit conceptual models of the documents they describe. It would, therefore, be a relatively straightforward task to translate these conceptual models into the data model used in a conceptual framework model for CIC. This would mean that the information contained in documents such as procurement orders, written specifications and contracts could be related to product model or activity model information in explicit ways useful for automatic retrieval of information.

5.4 *Product models*

A typical use of conceptual modelling techniques in construction IT research and development has been for structuring the information about the design artifact itself (product data models). Since a building contains a large number of greatly varying physical elements, a large number of diversified object classes are needed for the definition of a building product data model.

In the following, a building product data model defined in Finland (the RATAS model [31, 32]) is used as an example of an embryonic building product model. The underlying data model is the entity relationship model enhanced with inheritance of properties between classes.

A fundamental aspect of the model is the use of an abstraction hierarchy for decomposing a building. The abstraction levels chosen in the RATAS model are: building, system, subsystem, part and detail, as exemplified in Fig. 9. Objects from the higher levels in the abstraction hierarchy are especially useful in early design stages. 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 attributable data, which is defined in the detailed design phase.

The five abstraction levels chosen in the RATAS-model are not the only ones possible, but seem appropriate for describing a building functionally from the building designer's or the user's viewpoint. Alternative ways for organizing the abstraction hierarchy have been suggested in other research projects [33, 34]. Clearly the choice of abstraction levels depends on the viewpoint chosen (urban planning, building design, prefabrication). The same physical artifact can be the whole product in one viewpoint and a detail in the other.

An important feature of the entity-relationship data model used for defining the RATAS-model is the association of a number of attributes with each object. In order to describe the building as a product, we also need data structures for describing how different objects are interrelated, otherwise the building description would only be a computerized bill of quantities. Two generic types of relations are used for this purpose in the RATAS-model. The part of relation implements the aggregation/decomposition abstraction. Typically, it also implies an existency dependency between the objects. The other generic relationship type is the connected to relation, which could be subtyped into a large number of specific relationship types. (Danner has proposed a linguistics-based classification of different relationship types in building product models [35].)

The RATAS model is only one out of a number of embryonic building product models under development. The specific choice of this model to illustrate some typical features of product models is simply based on the author's familiarity with the model.

5.5 *Object-oriented activity modelling*

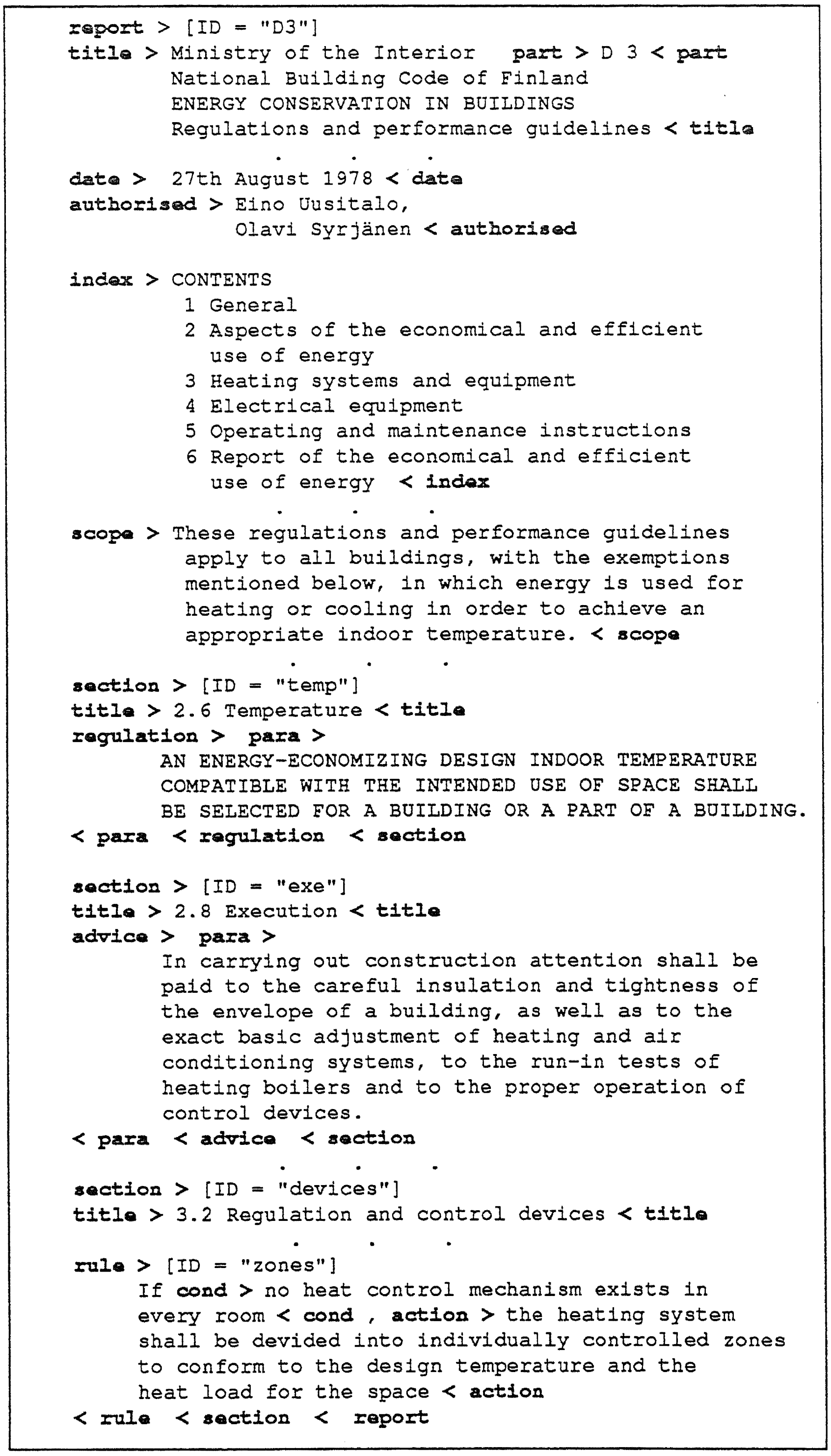
A different approach to modelling activities has been used in a number of research projects developing knowledge-based systems for construction management. In these, the use of object-oriented methods (in particular frame-based systems) for modelling activities provides the means for integration with product model building descriptions. Examples of prototype systems using this approach are Oarplan [36], Construction Planex [37] and Builder [38].

In all of these systems, activities are modelled as frames or semantic networks. This means that the data structuring mechanisms of these data models can be used for the modelling of:

- a decomposition hierarchy of activities;
- a specialization hierarchy of activities;
- relationships between activity objects and product model objects;
- temporal (precedence) relationships between activities.

All of these prototypes have, in fact, implicitly defined limited product data models representing the information needs of construction process management. These models contain mainly component level objects (columns, slabs, etc.). The description can also contain information about the spatial relationships of these components, since this information can be used for determining the temporal relationships between the corresponding activity objects.

An illustration of this type of approach to activity modelling is given in Fig. 10. The system in question, ATOP, was developed using expert system and hypermedia software and is used for generating and managing construction schedules [39].



```
report > [ID = "D3"]
title > Ministry of the Interior   part > D 3 < part
        National Building Code of Finland
        ENERGY CONSERVATION IN BUILDINGS
        Regulations and performance guidelines < title
                    .    .    .
date >  27th August 1978 < date
authorised > Eino Uusitalo,
             Olavi Syrjänen < authorised

index > CONTENTS
         1 General
         2 Aspects of the economical and efficient
           use of energy
         3 Heating systems and equipment
         4 Electrical equipment
         5 Operating and maintenance instructions
         6 Report of the economical and efficient
           use of energy  < index
                    .    .    .
scope > These regulations and performance guidelines
         apply to all buildings, with the exemptions
         mentioned below, in which energy is used for
         heating or cooling in order to achieve an
         appropriate indoor temperature. < scope
                     .    .    .
section > [ID = "temp"]
title > 2.6 Temperature < title
regulation >  para >
       AN ENERGY-ECONOMIZING DESIGN INDOOR TEMPERATURE
       COMPATIBLE WITH THE INTENDED USE OF SPACE SHALL
       BE SELECTED FOR A BUILDING OR A PART OF A BUILDING.
< para  < regulation  < section

section > [ID = "exe"]
title > 2.8 Execution < title
advice >  para >
       In carrying out construction attention shall be
       paid to the careful insulation and tightness of
       the envelope of a building, as well as to the
       exact basic adjustment of heating and air
       conditioning systems, to the run-in tests of
       heating boilers and to the proper operation of
       control devices.
< para  < advice  < section
                     .    .    .
section > [ID = "devices"]
title > 3.2 Regulation and control devices < title
                     .    .    .
rule > [ID = "zones"]
     If cond > no heat control mechanism exists in
     every room < cond , action > the heating system
     shall be devided into individually controlled zones
     to conform to the design temperature and the
     heat load for the space < action
< rule  < section  <  report
```

Fig. 8. The Standard Generalized Mark Up Language (SGML) can be used for structuring documents in an application-independent way. The example is taken from the Finnish building regulations.

6. BASIC COMPONENTS OF A FRAMEWORK MODEL

6.1 *Overall modelling strategy*

In the sections above, examples of different modelling approaches; graphical activity modelling, data flow modelling, digital documents modelling and conceptual modelling were presented. In the following we will try to show how the conceptual data modelling approach could be extended to cover the majority of information generated during a construction project. Such information can reside in CAD-drawings, in time-tables, cost estimates, contracts and other type of currently used documents.

Such an overall framework conceptual model would in the first instance, have to be defined on a highly abstract level. It is assumed that this framework could later be

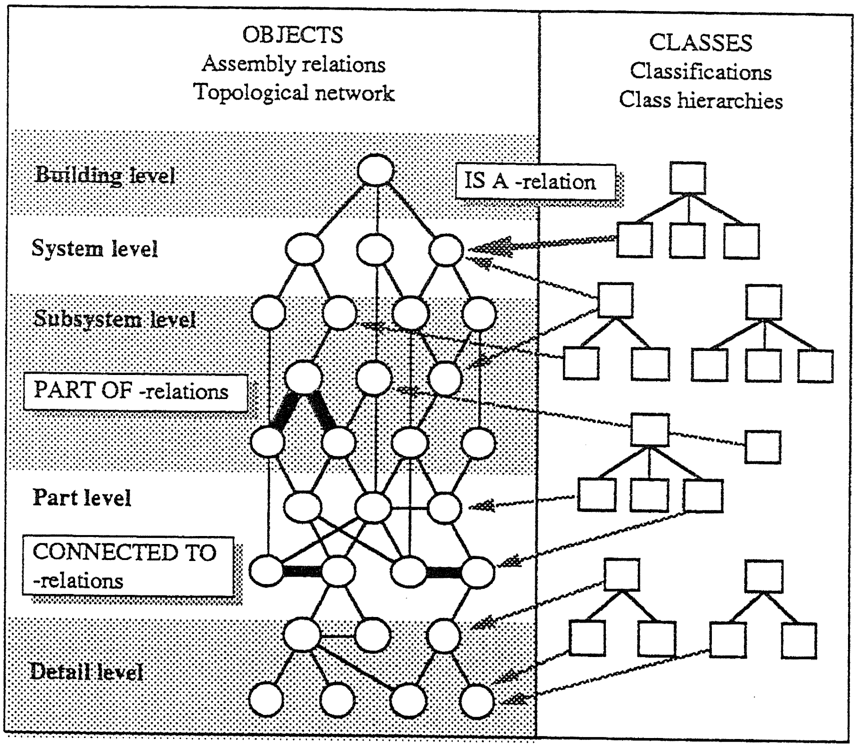


Fig. 9. In the RATAS-model, a building is described by objects forming a decomposition hierarchy with five levels.

detailed using both decomposition and specialization abstractions to the desired level of detail. This task would be of the same magnitude as the definition of a complete building product data model, something that is very time- and resource-consuming. Serious objections could no doubt also be raised as to whether such a task is at all realistic, or even desirable.

A more realistic scenario is for a framework conceptual model to be used as a basis for the development and subsequent integration of partial standards; for instance application protocols of STEP in the product model domain, EDI messages in the trade sector, traditional building classification systems, data bases of materials and resources.

In the following, suggestions for some very generic objects classes will be presented. The data model used is the data model underlying the IDEF1X methodology which is very close to the Entity-relationship model [22] enhanced with the subtyping abstraction. The emphasis is more on identifying a dozen or so generic object classes than on modelling their detailed interrelationships. The list of object classes is probably not comprehensive. Attributes are virtually not defined at all.

6.2 *The construction process entity*

First of all, we need to define a generic object class to which all the objects in the conceptual model belong. Let us call this fundamental class 'construction process entity'. Due to the inheritance mechanism used, all other object classes are subclasses of this class. On this level a unique identifier is specific for each object. Consequently no two objects belonging to the same construction project can have the same identifier. This identifier should be used throughout the whole construction lifecycle, and can be computer-generated. An analogy from administrative systems is a person's social security number. Data retrieval and updating in the case of changes would be much easier if such an identifier were to be universally used in a CIC environment.

Other types of information which could be defined on this very generic level would be data on the creation date for the object in question, versions, author, etc.

One starting point for defining subtypes of this 'root' class is to look for the major classification categories in use today in building classification systems. The draft standard under development by an ISO working group for the classification of information in the construction process, contains a good overview of the scope and information content of traditional classification systems [40]. Another source of potential classes is to look for data implicitly contained in activity diagrams or other such models.

The generic description used in many national classification standards is based on the three categories: activities, results, resources. An activity uses resources to produce results. Traditional construction classification sysems often tend to equate results to buildings and their parts. This is due to a desire to distribute total construction costs over building parts, which is useful for cost analysis purposes. It is, however, evident that information (mostly delivered as documents) and services are other important subtypes of results.

The construction process entity can be divided into a number of subtypes. However, it should be emphasised

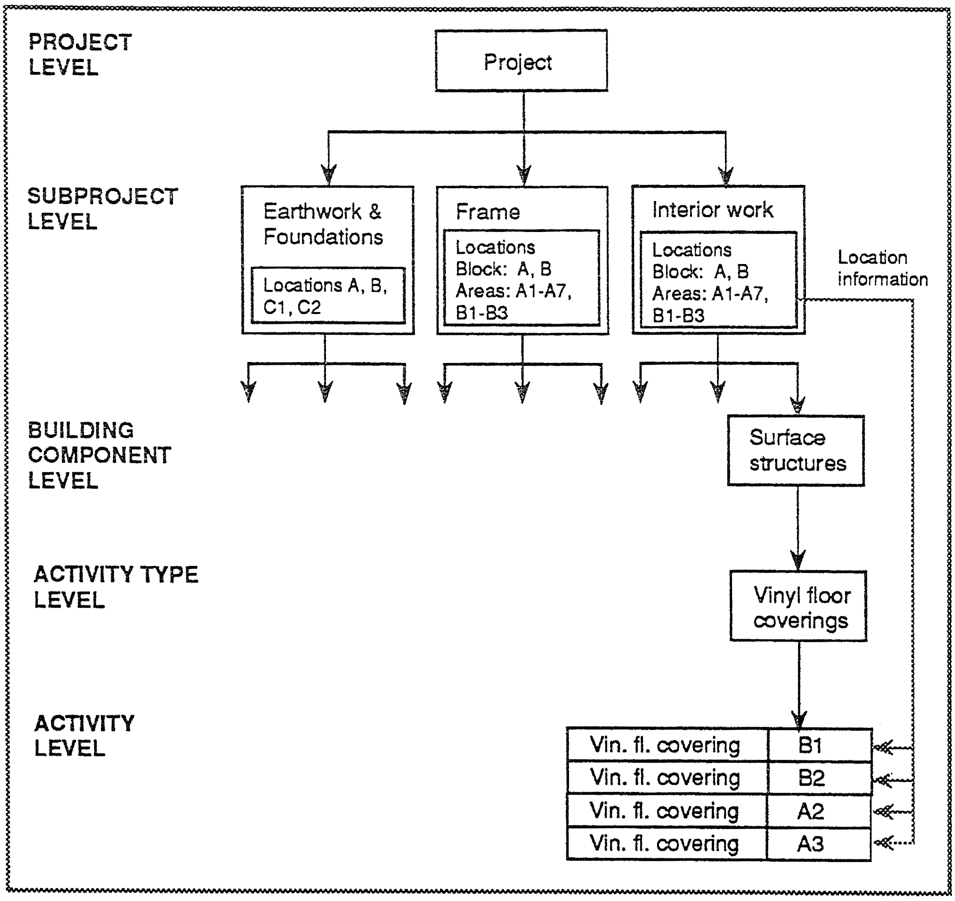


Fig. 10. Activities are treated as objects in a prototype software for the generation and management of construction schedules, forming their own decomposition hierarchy.

that the classification presented here is by no means the last word and more analysis is needed. The categories chosen should, nevertheless, illustrate the general idea. Subtypes are shown after their parent supertype using indentation.

- activity;
- result;
    - document;
    - physical object;
    - service;
- agent;
    - organization;
    - person;
    - machine;
    - facility;
- resource use;
- cost;
- contract.

The interrelationships between some of these concepts is shown in the IDEF1X diagram in Fig. 11.

Similar types of very general classes have been defined in other proposed generic conceptual models. The Global AEC reference model [41] contains the product definition unit as its basic class (entity). In follow-up work researchers from the Dutch research institute TNO have suggested expanding the original product model to also include activities and resources (Building Project Model [42]), and have called the corresponding generic entity 'Manufacturing Definition Unit'. In the Building Project Model, a Manufacturing Definition Unit can be further subtyped into activity, product definition and resource entities.

Böhms CIM reference architecture [21] contains as its fundamental entity the Manufacturing System Definition Unit (MSDU). This is in fact a cluster of three different types of entities: a manufacturing activity, a flow object and a CIM component, all related to each other by relationships.

6.3 *The activity entity*

As indicated by the presentation of object-oriented activity models in Section 5, a generic object class is needed for all kinds of activities in a construction project. This class should encompass all activities regardless of performing agent (for instance inspection for compliance with building regulations by a public authority should be included) or the type of activity (information processing, physical work).

Activity objects are at the very kernel of the framework model since they have relationships with many of the other generic classes in the model. Activities would have relationships with the results they produce, the resources they use and the agents that perform them. Another

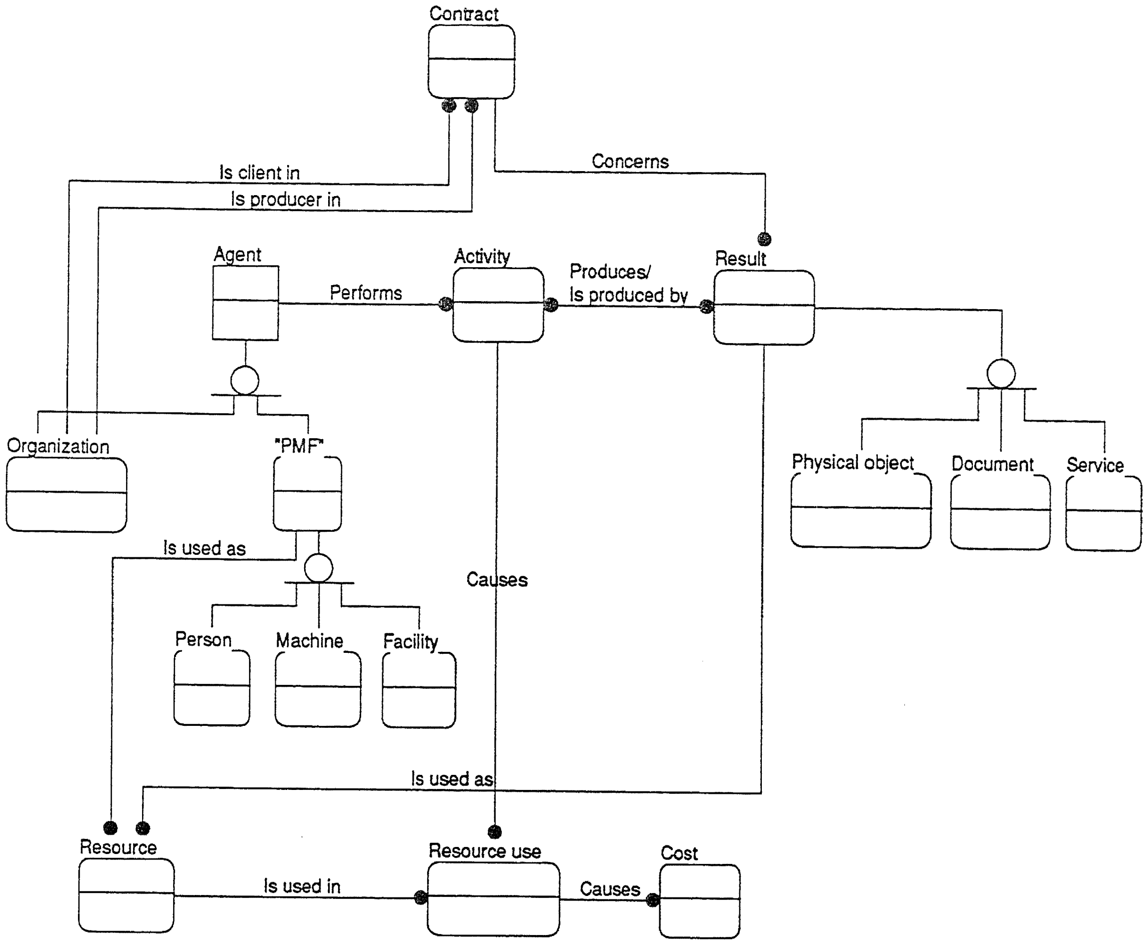


Fig. 11. This IDEF1X diagram shows the main generic object classes needed for the conceptual modelling of construction process information and some of their internal relationships. All the classes shown are subclasses of the "root" class, construction process entity.

important relationship type is with other activities, for instance specifying decomposition or temporal dependency relationships.

An important thing to note is that we could semi-automatically generate an activity diagram from an object-oriented activity description formulated by using the framework conceptual model. The SADT-diagram would be formed in two phases. In the first phase we would extract a subset of the overall activity model as a database view which only supports the data structures directly supported by SADT's underlying conceptual model. In the second phase, the corresponding graphical report, the activity diagram, would be generated. The process could also be reversed and the diagram could be used as an input medium for some data about a particular project's activities. Equally we could create many other kinds of graphic or non-graphic reports.

In a sense the relation between an activity-diagram and the underlying conceptual model is analogous to the relation between a construction drawing and the product data model design description underlying it. Both activity diagrams and construction drawings are used in practice because they are user friendly ways of presenting information for human interpretation and manipulation.

6.4 *The result entity*

The result entity is an example of an entity type needed for classification purposes, but which is intermediate in nature since most of the relevant information about results will be defined in the class descriptions of the subtypes of results. One reason for the inclusion of this class is the fact that on a generic level every activity will have one or many results and that a contract usually specifies a result to be produced.

6.5 *The document entity*

Data generated during a construction project is usually arranged in the form of documents. A document can collect and arrange information about a large number of microlevel objects, such as design objects, activities, resource use, etc. Documents also form decomposition hierarchies of their own, for instance the documentation needed for obtaining a building permit can consist of a number of lower level documents. A large number of document types, as well as procedures for identifying and numbering them, are defined by industry standards and guidelines issued by professional associations.

In the traditional construction process, documents can be subdivided into a number of well known, and reasonably standardized types such as drawings, specifications, bills of quantities, analysis results, contract documents (note the division between a contract as a conceptual object and the document describing it), design meeting minutes, etc. These can be subclasses of the generic document class. Unstructured sketches and personal notes could also be included in the document class since managing this type of information could be helpful in some applications—for instance in design tools. New types of documents which need additional defining information are EDI-messages, CAD-drawings, spreadsheet tables and E-mail messages.

Information retrieval in computer integrated construction necessitates macro-level information concerning the type, status and physical storage of information as well as the actual queries which, on a microlevel, extract the information needed once the relevant document has been found. For computer-generated documents, the software used and its version is also an important data item which needs to be stored in the document entity in order to support easy retrieval and conversion procedures. Databases pose special problems since they keep changing all the time. One way of dealing with this problem is to consider the states of the data base at particular times, which are communicated or open to queries for different parties in the project, as separate documents or versions of the same document.

An important feature is that each document must have an identifier which is unique within a particular project. The identifier attribute is in fact inherited from the superclass construction process entity. Existing practices for numbering drawings etc. could also be used for providing alternative identifiers easier to understand for human users.

Since documents can concern any kind of information, all the subclasses of the project information class can be used internally in the documents. Documents can, for instance, contain information about other documents.

The conceptual modelling of project documents seems to offer a lot of advantages. Using a conceptual model for documents would also enable us to model the relationships between these documents, and between documents and the overall model, in a way which would support advanced data retrievals. The difficult question of design version management could also be addressed within this submodel.

The submodel for documents could provide a good starting point for the development of database or hypermedia applications for project document management, in the form of a software independent specification for developing the kind of capabilities included in operating systems and user interfaces such as the Macintosh Multifinder or Windows 3. In such systems general information about files is easily accessible, searches to find specific files can be made easily etc. A hypermedia/database interface for construction project documentation would function in a distributed network environment, providing access to documents physically stored in different computers. Applications of this type would also have enough intelligence to load the software needed for the manipulation or conversion of the document automatically. Figure 12 shows a hypothetical screen image from such an application.

To each particular document subclass we could associate the conceptual non-document object classes which are utilized internally. Such work is already being undertaken for the specific case of drawings within the STEP-standardization process [27]. An interesting point is that different documents could contain information about the same object (for instance a building part, a specific activity). The link between these would be the identifier of the object, which should be unique within the construction project and not only within the document. If an object description is changed, it should then be possible to find all the references to it in all project documents by using the identifier.

An important aspect of document objects is that it would become possible to trace all other documents (including versioning information) which have been used as input information to the current document. If this relationship is stored bi-directionally (also in the document which has been used) it should be possible to manage changes in designs and plans and their propagation through using knowledge-based techniques.

A conceptual model of project documentation also lends itself to the description of current construction practice and works independently of the conceptual models of the other construction process entities. Consequently such a model will be used as a basis for management systems for project documentation based on current manual or computer-aided drafting techniques. This is of strategical importance since direct data retrieval from product models and databases covering other aspects of the overall project is only a distant dream at this time. Another factor to be taken into account is a major task still ahead of us: namely the digital management of the vast amount of documentation concerning existing buildings.

6.6 *The physical object entity*

A physical object entity describes any physical object with shape and location. (Documents are not included here, except in special cases such as books in library design where the physical space they occupy is a major design variable.) The location need not be static and can change with time. Also the shape may be dynamic, for instance for certain raw materials and fluids or gases. Physical objects can usually be modelled through 3D-models.

The data structures needed to define the end product itself and its parts would be the corresponding ones used in building product models. Since there is a lot of literature available on this subject and since the RATAS model was introduced earlier in this paper, it is not further elaborated here. Obviously, physical object entities and their subclass physical site objects could use the conceptual models developed within the STEP standardization effort for the description of shape, location, tolerances, etc.

It is, however, proposed that the physical object entity would also encompass the technical description of all kinds of materials, prefabricated building parts etc. which are primarily produced in factories and delivered and installed on site. In traditional classification systems these are often referred to as 'resources'. We propose a clear distinction between the technical description of a physical

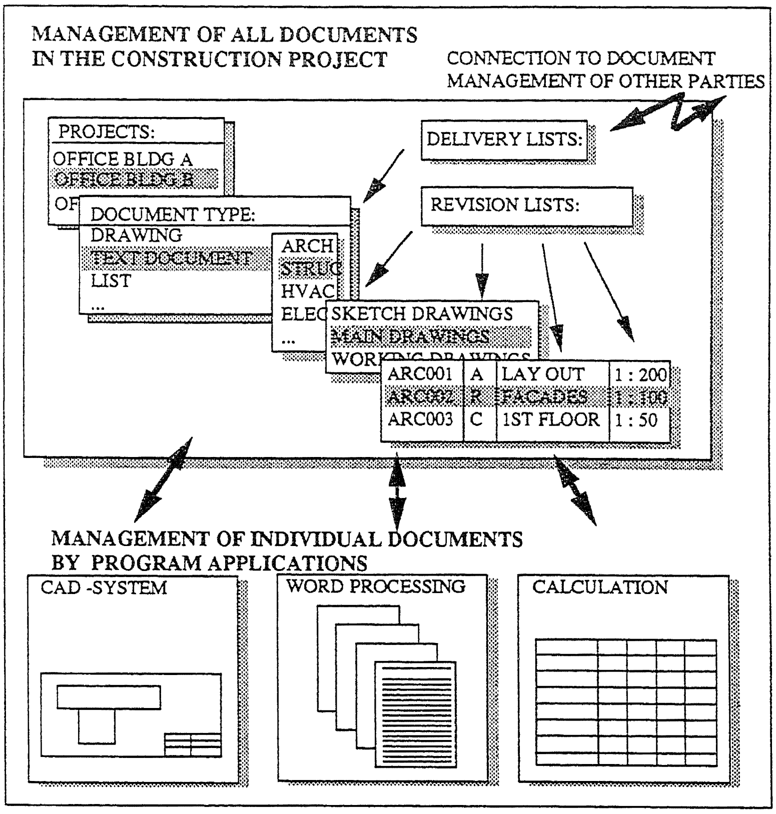


Fig. 12. A conceptual model of project documentation, including decomposition and subtype abstractions, would provide a good basis for developing distributed data base management systems for documents.

object and the description of its consumption ('resource use') in a particular activity.

Each building material or prefabricated part is in itself the subject of a manufacturing process which could be described in a specific model. Thus the distinction between the objects classified as physical site entities or building material objects is very slim. The distinction is, in fact, that materials have no static location on site. They can, however, be given dynamic location information in logistics applications. For logistics purposes it would also prove very useful to use aggregation objects containing information about materials bundled together for transportation or storage.

Very often materials are described in guidebooks or databases by type objects. Reference to them can then be made by specifying the unique code of this type.

The treatment of information about demolition work and temporary structures on site has been a major difficulty in many current building classification systems. In order to deal with this we need a generic entity for all object classes which have a static or dynamic (time-dependent) location on the construction site. Objects which can occupy a certain location should also obviously have shape information.

This generic class is very useful in order to deal with the objects which may be needed for the dynamic simulation of activities on the actual building site, for instance for constructability analysis or as visualizations for project management. All objects in the building product model automatically belong to this class. Other important categories of physical site objects are temporary structures during construction work (for example scaffolding), construction machinery and building materials during the time they are stored or used on site, existing structures which need to be demolished and so on. In the treatment of some of these classes, multiple inheritance may be needed. A machine, for instance, could at the same time belong to the agent class and the physical site object class, inheriting relevant attributes from both classes.

Objects belonging to the class describing the situation before construction starts (soil layers, sewage systems, existing buildings, etc.) can potentially have relationships with demolition activities but cannot have relationships with erection activities at least within the scope of a typical new construction project. Temporary structures are both created by activities and demolished by activities. The structures which today are described in building product models can be erected or be inherited from existing structures, but cannot be demolished during the actual construction phase. A conceptual diagram of these physical site object and activity subclasses is shown in Fig. 13.

6.7 *The service entity*

Services are such results of activities which are not physical objects or documents (information). The result of the activity of guarding the site, for example, is that no

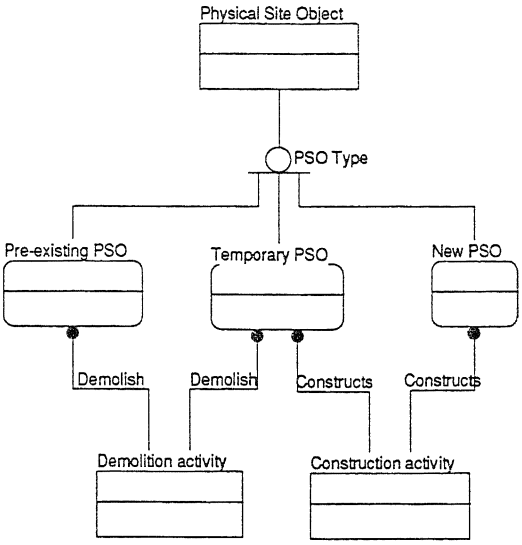


Fig. 13. Building product data models have so far concentrated on new construction. It is, however, easy to extend the conceptual modelling of physical site objects to also include temporary structures and structures which need to be demolished.

materials are stolen during the construction project and that the contractor saves more money than the service cost him.

6.8 *The agent entity*

An agent is any organization, person, machine, or facility which participates in the activities of the project. An agent performs some activity in the process. A computer application is an interesting new type of agent which could be included in the conceptual model. A fundamental aspect of an agent which distinguishes him from a product is that he/it has an existence outside the project and usually participates in several projects simultaneously or in sequence. Information about agents are typically included in many information systems of the parties involved in a construction project, for instance in the administrative systems of construction companies. Agents form decomposition hierarchies of their own (departments, project groups, work gangs), which is a useful feature for project management.

Machines can be described in ways which are quite close to the descriptions used in product models. More emphasis will, however, be laid on the formal modelling of the dynamic behaviour than in the case of static structures. This is for example quite obvious in the case of construction robots.

6.9 *Resource use and cost entities*

The resource use entity might be slightly difficult to explain. Each activity in the project demands a number of inputs in the form of resources which are 'consumed' or used. The actual use may be measured in manhours, tons, squaremeters, CPU-seconds, etc. A clear distinction is made between resource use, and the resource entities themselves, which may be documents, materials, machines or persons. The characteristics of these should be modelled in the corresponding parts of the framework model.

The use of any resource involves a cost, which in most cases could be measured by the amount of the resource consumed, but in some cases by the opportunity cost of that resource. The opportunity cost measures the customers willingness to pay for the resource in question. Information and land, for instance, are not actually consumed in production processes but a price can be charged for their use. This is one reason for distinguishing between cost entities and resource consumption entities.

One resource which is used in the construction process and which has a cost is money. Money is a resource used for activities such as pay salaries and bills. Interest on money is a cost item which can be quite significant and affects the overall construction cost from the contractor's viewpoint.

6.10 *The contract entity*

These types of objects provide the framework for describing the overall organization of the project. Contracts can be both large overall contracts or even small orders for materials. Typically, a contract entity would have links to agent entities, activities, result entities, document entities, etc. Cash flow information corresponds to the actual flow of money in the project. Methods such as EDI-messages are today already being applied to this category of information. The document type definitions of the SGML standard could also be used for implementing conceptual models of contracts etc.

Contracts form an interesting abstraction hierarchy. At the top we find the main contract for the whole project and further down we find subcontracts, material orders, etc. The important thing to note is that the contractual organization of the project can be modelled fairly independently of the modelling of project activities. (This is true as long as the choice of subcontractors doesn't influence the actual activities being undertaken.) This would enable us to simulate the same project using different contractual organizations.

6.11 *Additional generic object classes*

Additional object classes which could be included in the framework model are building performance and type objects.

Building performance entities are related to the modelling of the building over time. They may include equation systems describing the thermal and structural behavior of the building, the flow of people in the building during its use etc. Related to the description of the building in use are activities, resource use and contracts, much in the same way as in the actual design and construction process. All of these may in fact be modelled as activities which may or may not result in costs. The overall organizing activity is 'using the building' which can be considered the aggregate entity containing all the socio-economic and physical entities needed to model its behavior.

Research work which uses a uniform object-oriented approach for data and software management in energy-related building simulation [43], is already in progress. In a project carried out by the Technical Research Centre of Finland, a conceptual model integrating a product model of a radiator heating network with the objects representing the matrices needed to dimension the network has been defined [20].

In addition to instanced objects, type objects should also be included in the conceptual model. A type object is a simple or complex object which in addition to a conceptual schema contains actual attribute values for many of its attributes. Type objects are often described in manuals and general data bases, and can be instanced in creating the actual database descriptions of the project at hand. The notion of prototypes, which has been proposed in the theoretical literature is related to type objects [44]. The generic and specific objects of the GARM model are also examples of specialization hierarchies of type objects [41]. In the following some examples of type objects are given:

- Standard joint types for steel structures;
- Breakdowns of particular building elements into average resources and activities based on empirical data;
- Models of the stages of the building project;
- Models of the work sequences of robots for performing specific tasks.

The inclusion of type objects is currently of special importance in product model research [45], but the same generic class is also applicable to activities and other classes.

7. EVOLUTION OF INFORMATION THROUGH THE PROCESS LIFE-CYCLE

An aspect of the framework model which would need to be elaborated is the degree of determination of the information. Information about activities or building components can be planned or based on measurements of the actual components and phenomena. The further we proceed in a project's life-cycle, the more precise information tends to become. A lot of emphasis has, for instance, been given to this aspect in the General AEC Reference Model [41]. At the moment, the issue of how to model functional requirements for components versus actual physical characteristics, is also the subject for study also in other projects [46].

A starting point for an analysis could be the actual construction stage of the process. This means, for example, that we consider activities as they actually occurred, and not as they were planned. We can thus insert actual starting and completion data in activity objects (we postulate that we could actually measure these accurately), use empiric book-keeping data on the use of resources and describe the end result with a detailed 'as built' product model description. Bills and procurement orders also correspond to real documents in our database description. We could name the information structure of the object classes at this state 'construction process information after the fact'.

From this very voluminous and detailed set of object descriptions we can work backwards through several layers of abstraction to model the information content of preliminary design descriptions of the building, early activity schedules, cost estimates, etc. The information structures used in these are obviously related to the after-the-fact structures through abstraction relationships, which could be labeled; 'is estimate of'. As a rule the descriptions of planned objects is on a less detailed level than the description after the fact.

Our hypothesis is that any information structure that is included in earlier design and planning documentation can also be included in the after-the-fact description of the building since the information it contains is still valid. Note that we talk of information structure, not necessarily of the actual values of attributes etc. which of course may change. A few examples are given below:

- In earlier design phases decisions have been made about the U-values of walls. U-values are obviously still relevant in the detailed description of the finished building.
- We might apply database information about the cost per area of a certain type of wall structure, or about the cost of certain types of spaces in order to obtain a cost estimate of a rough early design. Actual building cost could also be accumulated in this way (this is in fact what is done in the empirical work which is needed to update such databases).

Another point that should be made is that in actual database descriptions of buildings we do not need to specify actual values for all the information in this complete conceptual model. During most of the stages of the process the information is incomplete.

The conceptual separation of the different stages in the utilization of these data structures could be achieved in several ways:

- By using different data structures (object classes) for structuring the information about the same physical or abstract object during different phases of the process.
- By tagging all information (either at the object, relationship or attribute level), with information about the project phase (activity) during which the information has been created.
- By using the same conceptual data structure throughout, but by applying classification according to project phases to the documents in which the information is delivered.
- By using the same conceptual data structure throughout, but using different data base views to retrieve the information which is delivered during different design, planning and monitoring activities.

The last two alternatives seem particularly interesting. We would start with a large coherent conceptual model which would mainly be based on a detailed analysis of the situation after the fact. We would then take the information in current documents used in earlier design and planning stages and see how they map against the overall conceptual model. This would give a lot of flexibility because different mappings could be used for different design and construction process practices.

The data base view alternative would also be quite advantageous. A methodology which might be useful is the application protocol concept [47], which is going to be used in STEP to model particular data needs of narrow application areas against the overall model in STEP.

8. A SMALL EXERCISE ILLUSTRATING THE APPROACH

The approach presented in this paper was utilized in a small theoretical exercise modelling the detailed cost estimation of a specific type of internal partition wall [48]. The results of the exercise are shown in Fig. 14. In the figure, subcomponents of a parent object are presented in smaller boxes below the aggregate object. The way of handling information is based on traditional methods already in use, but the information has been restructured using the above framework model.

The starting point is a product model description of the partition wall, possibly defined by the architect using a library of type objects. The description includes information about the subcomponents of the partition (frame, board, etc.) and about the amounts of these included in the final product.

The actual use of material resources in erecting the wall doesn't coincide with the above amount, due to losses during production, cutting of boards and other such reasons. Information about average losses as well as the manpower time needed per unit of wall of this type can be found in the type activity object for Gypsum board installation. Such type objects can be found in company guidelines or in manuals provided by general construction information services.

Using the product model description and the type activity object, the estimated resource use can be calculated as a complex resource use object. In its subobjects the resource use is still in noncompatible formats (m2, m, hours). In the databases containing type objects describing materials and different professional work trades, we find the cost per unit of resource used, and the total cost of erecting the partition wall can thus be calculated.

Calculations such as this are already performed today by the quantity surveyors or estimators of construction companies. In many cases they already have computing applications to help them. These are, however, usually

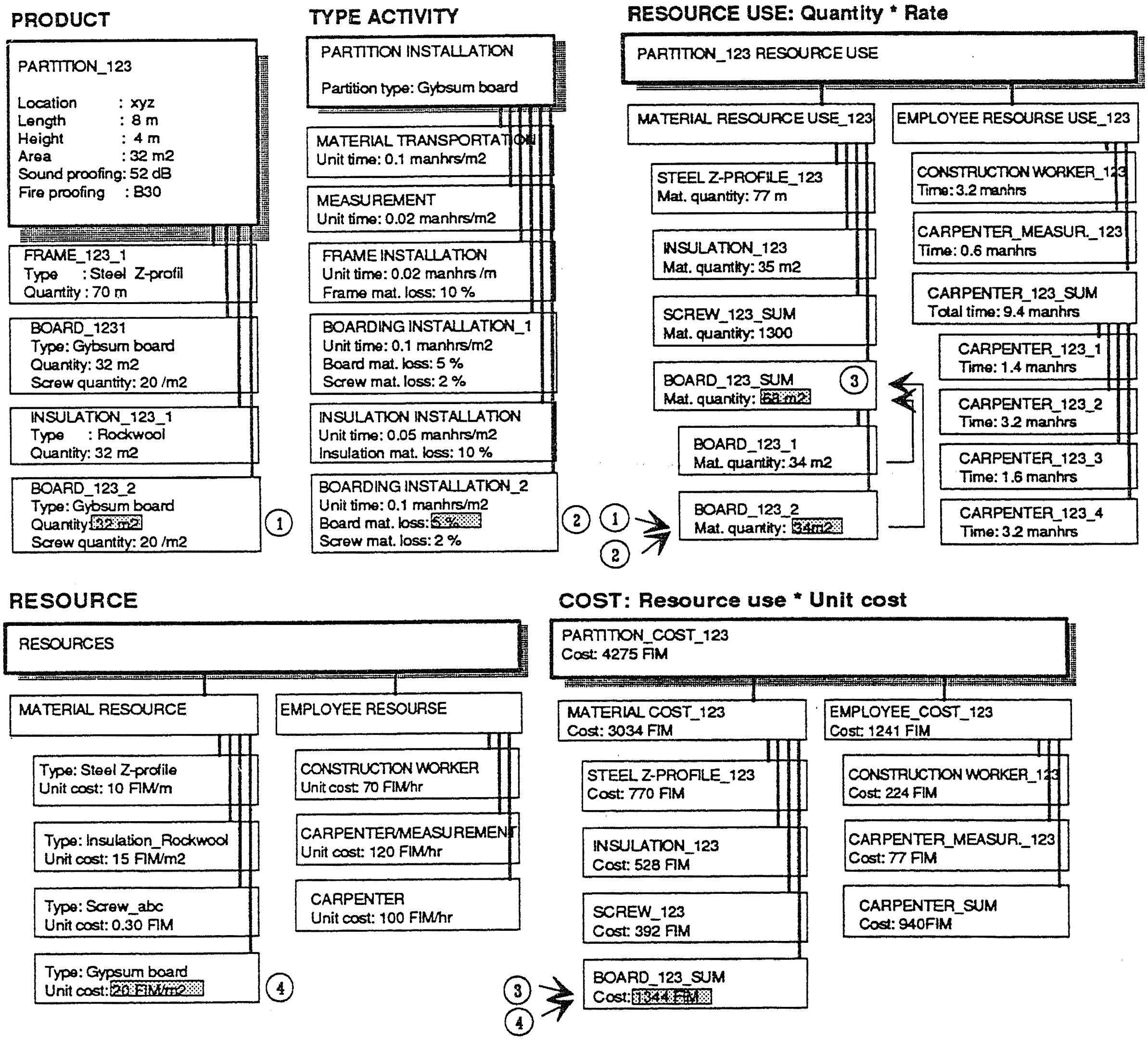


Fig. 14. A small paper exercise was conducted to test the approach on the detailed cost estimation of partition walls. The numbered circles indicate how information from different conceptual object classes is combined to produce the final cost estimate.

custom-made closed systems and necessitate manual input of data.

In the approach we present. the information is split into relatively independent databases which can be updated separately and from which information can be extracted for many other purposes. The integration of the information will be achieved by intelligent data base queries utilizing the classes and data structures defined in the framework model. In the following a few examples are given of possible hypothetical queries in such an environment.

- Find all the contracts for the activities following an activity which has been delayed and notify the parties involved of the currently planned timetable for their activities.
- Obtain the basic data for simulating the project, using alternative construction techniques or different contractual organizations.
- Obtain the information about the site, the building objects and the activities needed as input data to perform a 3-D simulation over project time, in order to visualize the progress of work and test constructability.
- Obtain the information concerning all other documents which contain partly shared information with a document which has been changed and notify the issuers or users of those documents about the possible changes.

9. CONCLUSIONS

A generic framework for describing construction information has been suggested in this paper. Since this paper only attempts to point out the directions for further research, no attempts are made to pursue the integration effort into a cohesive, comprehensive model. It is sufficient to point out that the use of a unified data modelling methodology, which covers all types of information generated and used during a construction project, and which specifically allows the free definition of semantically meaningful relationships between any object classes, provides the tools for building such a comprehensive model. In earlier research a number of different modelling techniques have been used for modelling more limited information domains resulting in partial models of different aspects of the overall information management in a construction project. Such current models are very difficult to integrate.

The major contribution of this paper is the proposal that conceptual modelling techniques based on some suitable data model should be used for modelling all kinds of construction process information. The specific inclusion of document, resource use and contract objects in such a framework will hopefully result in an extension in scope of current research efforts which seem to focus on building product models and knowledge-based process planning.

The usefulness of this approach depends on many factors. The first prerequisite is obviously the definition of comprehensive conceptual models of construction project information and their standardization at a national or international level. This is quite similar in scope to the work on traditional building classification systems and similar in methodology to research on building product data models.

The second prerequisite is that such conceptual models, or parts of them, are used as a basis for the development of a large variety of computer applications all over the construction industry. Alternatively conversion programs to and from neutral transfer formats based on such conceptual models have to be developed. Such work has already been done in the area of CAD data transfer, but the exchange of information between heterogeneous applications has hardly been discussed.

If these conditions are fulfilled, this approach has a lot to offer. Specifically, it allows different computer applications a much more straightforward and automated approach to information produced by other applications. On a more theoretical level, this approach could provide a comprehensive framework where existing traditional building classification systems could find their places as subclassifications of the generic object classes.

A lot of research needs to be done in this area. Hopefully this paper has shown some possibilities offered by the use of conceptual modelling techniques for integrating the research work currently being done in domains such as as building classification systems, CAD data exchange, product data models, knowledge-based project management and EDI, to mention a few.

Acknowledgements—The ideas presented in this paper were conceived during the final stages of two research projects, one focusing on building product data models, the other on overall strategies for computer integrated construction. Both projects were part of a research programme for Information and Automation Systems in Construction carried out by the Technical Research Centre of Finland during the years 1988–91. The author is especially indebted to Kari Karstila for his encouraging comments and criticism, as well as for carrying out the exercise presented in Section 8. Matti Hannus also deserves thanks for participating in many inspiring conceptual modelling sessions.

REFERENCES

1. D. Vanier and M. W. Grabinsky, Integrated Computer-Aided Building Design Systems, *AREC-DAO87 Conference Proceedings*, *ITEC*, *Barcelona*, pp. 34–37 (1987).
2. V.-P. Saarnivaara, Information Technology as a Tool for Industrialized Construction Systems and Processes, *2nd Finnish-French Colloquium for Information Technology in Construction*, *VTT Symposium* 119, pp. 10–30 (1990).
3. R. Wright, Computer Integrated Construction, *IABSE Proceedings* P-123/88, pp. 17–25, *IABSE Periodica* 1/1988.
4. H. C. Howard, R. E. Levitt, B. C. Paulson, J. G. Ohl and C. B. Tatum, Computer Integration: Reducing Fragmentation in the AEC Industry, *J. Comput. Civil Engng* 3, 18–32 (1989).
5. W. Gielingh, General AEC Reference Model (GARM), *Proceedings of the Seminar 'Conceptual Modelling of Buildings'*, CIB working commissions W74 and W78, Lund, Sweden, Svensk Byggtjänst, 14p. (1988).

6. V. E. Sanvido. An Integrated Building Process Model. Technical Report No. 1, Computer Integrated Construction Research Program. Dept. of Architectural Engineering. Pennsylvania State University. U.S.A. (1990).
7. BIM Bouw Informatie Model. Version 3.1. IOP-Bouw. Rotterdam. 2 folders (1989).
8. Structuration des donnees communes pour la gestion technique du patrimoine. Cahier du C.I.B. Nr. 6/1990. Club Informatique et Batiment. Paris. 32 p.
9. A Practice Manual on the Use of SfB. CIB Publication 55. An Foras Forbartha. Dublin. 122 p. (1986).
10. Y. E. Kalay, ed., *Computability of Design*. John Wiley. New York (1987).
11. R. D. Coyne. M. A. Rosenaman. A. D. Radford. M. Balachandran and J. S. Gero. *Knowledge-Based Design Systems*. Addison-Wesley. Reading. Mass. (1990).
12. M. Brodie. J. Myopoulos and J. Schmidt (eds). *On conceptual modelling—Perspectives from artificial intelligence, databases and programming languages*. Springer. New York (1985).
13. J. Peckham and F. Maryanski. Semantic Data Models. *ACM Computing Surveys* **20**, 153–189 (1988).
14. R. Hull and R. King. Semantic Database Modeling: Survey. Application, and Research Issues. *ACM Computing Surveys* **19**, 201–260 (1988).
15. K. Dittrich. Object-Oriented Database Systems: the Next Miles of the Marathon. *Inf. Syst.* **15**, 161–167 (1990).
16. Industrial Automation Systems—Exchange of Product Model Data—Representation and Format Description, ISO TC184/SC4/WG1 Document N284, 607 p. (1988).
17. D. Schenck. Express, ISO TC184/SC4/WG1, Draft N 210, 77 p. (1988).
18. Information Modeling Manual IDEF1-Extended, D. Appleton Company Inc., report, 110 p. (1985).
19. J. Turner. The Conceptual Modeling of a Building, *Computer Integrated Construction* (newsletter), Vol. 1/1, Wix-McLelland publs., Thatcham, U.K. (1990).
20. R. Talonpoika and H. Rissanen, The Integration of a Radiator Network Dimensioning Program With a Building Product Model, Building Simulation '91 Conference Proceedings, Sophia-Antipolis, France, IBPSA, 20–22.8.1991.
21. H. M. Böhms, Reference Models for Industrial Automation, Doctoral thesis, University of Delft, Netherlands, 150 p. (1991).
22. P. Chen, The entity-relationship: Towards a unified view of data, *ACM Transactions on Data Base Systems* **1**(1), 9–36 (1976).
23. D. A. Marca and C. L. McGowan, SADT—Structured Analysis and Design Technique, McGraw-Hill, New York (1987).
24. T. DeMarco, *Structured Analysis and System Specification*, Prentice-Hall, Inc., Englewood Cliffs, N.J. (1978).
25. R. M. Rangan and R. E. Fulton, A Data Management Strategy to Control Design and Manufacturing Information, *Engng Comput.* 7, 63–78 (1991).
26. Information Interchange in the Tendering and Bidding Phase—Dataflow, BPS-publication 89, Part I, BPS-centret, Horsholm, Denmark (1991).
27. J. Fowler, Application Protocol for the Exchange of 2d Geometrically Explicit CAD Drawings with Explicit Annotation, CEN/CLC/AMT/WG/STEP Document N46, CADDETC, Leeds, U.K., 52 p. (1991).
28. A Solution to Your Information Management Need, The SGML Approach, Brochure, Aspen Systems Corporation and Integrated Microcomputer Systems, Inc., 9 p.
29. P. McKell, Intelligent documentation—from word processing to knowledge processing, *Expert System User*, pp. 30–32, September (1990).
30. P. Debras, P. Poyet and E. Brisson, Expert Systems and Documentary Databases, *Proceedings of the Workshop on Computers and Building Regulations, VTT Symposium 125*, Technical Research Centre of Finland, Espoo, Finland (1991).
31. B.-C. Björk, Basic Structure of a Building Product Model, *Computer-Aided Design*, **22**, 71–78 (1989).
32. B.-C. Björk and H. Penttilä, A Scenario for the Development and Implementation of a Building Product Model Standard, *Adv. Engng Software*, **11**, 176–187 (1989).
33. K. Law and M. Jouaneh, Data Modelling for Building Design, *Proceedings of the Fourth Conference on Computing in Civil Engineering*, ASCE, New York, pp. 21–36 (1986).
34. Neutral byggproduktmodell-systembeskrivning, Royal Building Board, Technical Bureau report 109, Stockholm, 200 p. (1990).
35. W. F. Danner, A Global Model for Building Project Information : Analysis of Conceptual Structures, Report NBSIR 88-3754, National Bureau of Standards, Washington D.C., 38 p. (1988).
36. A. Darwiche, R. Lewitt and B. Hayes Roth, OARPLAN : "Generating Project Plans in a Blackboard System by Reasoning about Objects, Actions and Resources, CIFE Technical Report No. 002, Stanford University, 31 p. (1989).
37. C. Zozaya-Gorostiza, C. Hendrickson and D. Rehak, Knowledge-Based Process Planning for Construction and Manufacturing, Academic Press, San Diego (1989).
38. Robert D. Logcher and D. Sriram, CAE techniques for distributed design, Proceedings of the CIB working group W55 conference on Building Economics and Construction Management, Sydney, CIB, Vol. 2, pp. 345–362 (1990).
39. Kalle Kähkönen and P. Laurikka, A Computer Program for Generating and Managing Schedules for Construction Projects, *2nd Finnish-French Colloquium for Information Technology in Construction, VTT Symposium* 119, pp. 140–152 (1990).
40. H. Karlsson and T. Allott, Classification of Information in the Construction Process, ISO TC59/SC13, Discussion paper, 25 p. (1990).
41. W. F. Gielingh, General AEC Reference Model, ISO TC 184/SC4/WG1 Document N329 (1989).
42. G. T. Luiten and F. P. Tolman, Design for Construction (DfC) in the Building and Construction Industries, *Preproceedings of the CIB W74 and W78 Conference "Computer-Integrated Construction"*, Tokyo, 17–19.9.1990.
43. J. A. Clarke, The Energy Kernel System, *Energy and Bldgs* **8**, 259–266 (1988).

44. J. S. Gero, M. L. Maher and W. Zwang, Chunking Structural Design Knowledge as Prototypes, in J. S. Gero, ed., *Artificial Intelligence in Engineering: Design*, CPM/Elsevier, Amsterdam, pp. 3–21 (1988).

45. B.-C. Björk, Building Product Data Modelling—Experiences of Prototype Development, Accepted paper, *Microcomputers in Civil Engineering*, to be published Spring 1992.

46. C. Eastman, A. Bond and S. Chase, A Formal Approach for Product Model Information, Report nr. 3, UCLA, Graduate School of Architecture and Urban Planning, Los Angeles, 37 p. (1989).

47. J. Fowler, STEP ENV-Framework, Draft Document CEN/CLC/AMT/WG STEP N13, CADDETC, Leeds, U.K., 11 p. (1990).

48. K. Karstila, B. C. Björk and M. Hannus, A Conceptual Framework for Design and Construction Information, *Proceedings of the First International Symposium on Building Systems Automation-Integration*, Madison, Wisconsin, U.S.A., 2–8.6.1991.