

# Integrated Construction Project Document Management (ICPDM)

B.-C. Björk , P. Huovila and S. Hult

Technical Research Centre of Finland, VTT

This is a restored version of a paper published in Behesti, M.; Zreik, K. eds. Advanced Technologies - architecture - planning - civil engineering, proceedings of the EuropIA'93 conference, Delft, 21-24.6.1993, Elsevier, Amsterdam, 1993, pp. 135-146.

The layout differs from the original and the appendix has been added (it was included in the submission but left out of the published paper due to size restrictions).

The emphasis in research concerning methods for computer integrated construction has recently been on advanced data base techniques (product models) and AI applications. Another type of computer-aid for integration which could have a significant impact on practice in a shorter time-frame focuses on the management of the different documents which are produced by the different participants in the construction process. A functional description of such an integrated construction project document management (ICPDM) system is presented in the paper.

Keywords: CIC, conceptual modelling, integration, document



## 1. INTRODUCTION

Computer integrated construction (CIC) is a high priority research area in many of the leading research units dealing with the use of IT in construction. Among the subjects of research are product models, knowledge-based construction management systems, concurrent engineering etc. [Dupagne 1991]. Looking at this research from the viewpoint of industry there seems to be too big a gap between the medium term needs of industry and the subject area of much of the research. Standardised product model descriptions of buildings may prove extremely useful in the long run, say within 10-15 years. Enterprises involved in projects where some of the participants produce drawings using CAD systems and others manually and where a few participants have email but the most common denominator is that almost everybody has a telefax, need less sophisticated methods for managing information, methods which could lead to improvements in management procedures and productivity within 2-3 years.

A different approach for integration is offered by concentrating on a less ambitious level, namely integrated document management. This is an area which by and large has been neglected by the construction IT research community but which is receiving increasing attention both by developers of commercial CAD software and by organisations developing EDIFACT-messages for construction. The former are starting to offer document management systems as add-ons to their CAD packages. The latter are starting to realise the need for such systems as the scope and number of EDIFACT-messages is increasing, for instance to cover the digital transfer of drawings [Neutebooms 92]. The rapid proliferation of data transfer networks is obviously a prerequisite for the emergence of such systems.

The approach which will be introduced in this paper concentrates on the management of documents in digital form, not on the management of the information within documents or databases, as exemplified by the product model approach or the internal data content of EDI-messages. In this approach, which we will call integrated construction project document management (ICPDM) the atoms which we deal with are individual documents issued during a construction project (contracts, cost estimates, drawings, bar charts, specifications etc.). The objective is to use computer technology to manage all this information in such a way as to allow quick access to the right information at the right time.

A more theoretic framework placing this approach in a larger context of levels of data transfer in CIC has been presented elsewhere [Björk 92b].

## 2. THE CONTEXT OF THE PROJECT

### 2.1 The CIC pilot project

The development of the ICPDM system is a subproject of a larger project ( the CIC pilot project) which aims at implementing some of the product modelling methods which have been developed during the RATAS project. The project is financed by the Technology

Development Centre of Finland and it involves several participants including VTT, the University of Technology of Tampere, a small architectural practice and a large HVAC & electrical design firm. A key feature of the project is the consistent use of the formalised modelling methods, EXPRESS and SADT, which will be presented later on in this paper.

## **2.2 The empirical test case**

The development of the ICPDM system is partly based on the analysis of documents from an empirical test case: a construction project that resulted in a two-storey office and exhibition building for car sales. The main criteria for the choice was to find a completed construction project with all documents available (not only drawings) and where as large a share of these as possible had been produced using computers.

The chosen documentation consists of more than 800 documents covering a variety of drawings, different types of lists, minutes, assembling instructions, schedules, reports, calculations, calls for tenders, tenders, contracts, invoices, delivery notes etc.

## **2.3 Scope of the ICPDM system**

The intended system covers all documents that are produced during a single construction project from inception through the project lifecycle. In this initial phase the emphasis is on the design and construction phases, but it is straightforward to extend the system to also cover the documentation of the maintenance phase. The aim of the system is to facilitate access to information regardless of whether the document has been produced using software or manually, and in the former case regardless of the type of originating software.

# **3. A HARDWARE AND SOFTWARE ARCHITECTURE FOR THE ICPDM SYSTEM**

## **3.1 Principles for defining the hardware architecture**

The design of the architecture for the ICPDM system builds on two principles. Firstly, the combination of project participants changes from one construction project to another. This means that switched networks should be used. Secondly, the site is the most important place to use the documents, but it can't use more advanced computer and telecommunication systems during the project. This means that the system must use a data transfer media that can be accessed through low-end computer systems, i.e. personal computers or small workstations equipped with modems.

The management program will be developed for personal computers (PC's) running Windows version 3.1. This configuration was selected because it is the de facto standard in the construction business today. A PC is flexible enough to be easily transferred from one site to another.

### 3.2 Separating documents and their management data

Using modems means that it is not sensible to send all the documents right away to all parties. Instead, it is sensible to limit the transfer to the document management data. That is why in the ICPDM system the document data itself and its management data are separated from each other. The management data will be sent to every party actively, that is, the creator of a document sends messages to other parties, but the data must be retrieved by the other parties from the file servers of the data transfer network (see fig. 1).

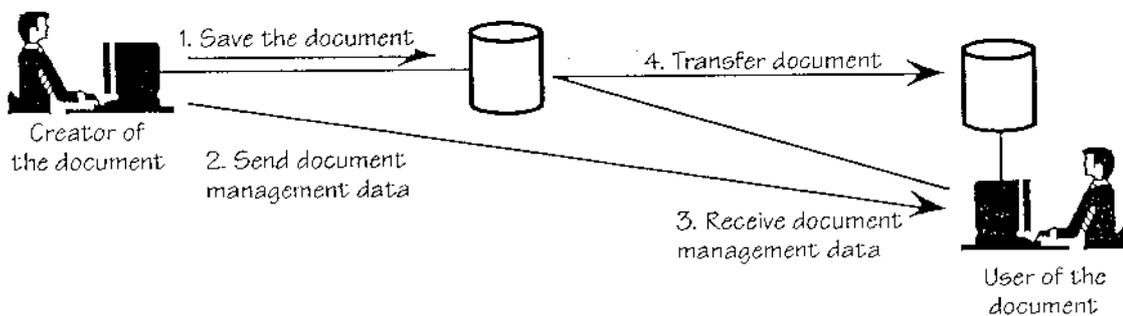


Figure 1. Active management data distribution, active document retrieval.

### 3.3 Software architecture

The ICPDM consists of local management programs running on each party's workstation and a message handling system provided by a telecommunication company. The file transfer can be arranged separately using another means of communication.

The media for transferring the document files should be fast enough. Systems today use modem connections through the standard telephone network with the speed of max. 9600 bps. This speed isn't fast enough, and one should consider using faster switched networks such as ISDN in the future.

### 3.4 User interface

The user interface will be developed in accordance with the standard Windows user interface. The typical user interface includes non-modal windows, dialogue windows, pull-down menus, buttons and fields. A hypothetical example screen is shown in figure 2.

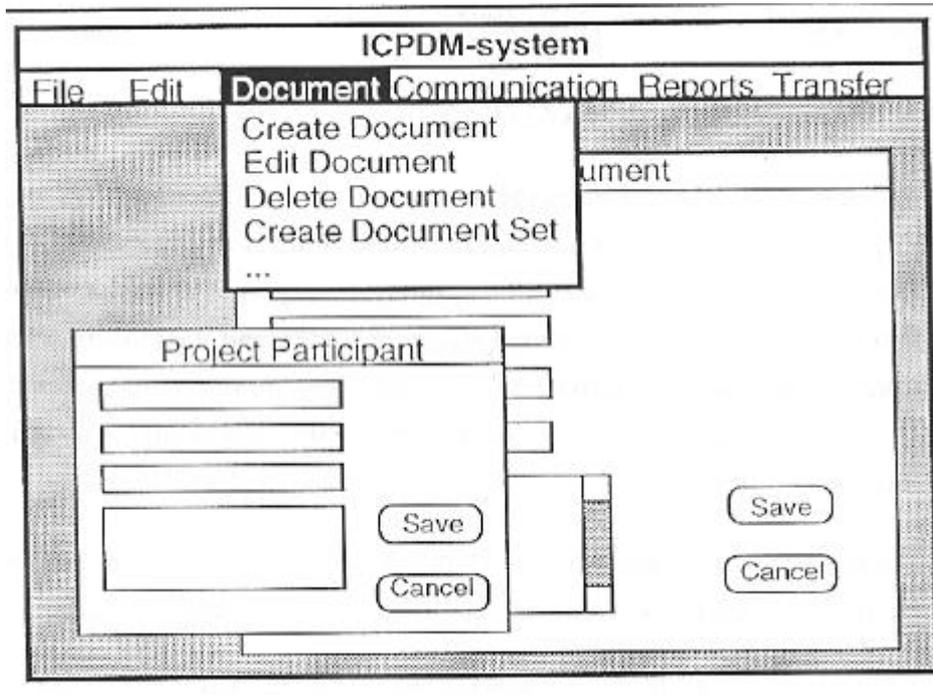


Figure 2. A sketch of a typical ICPDM system screen.

The management program will include the following functionality;

- creating, editing and deleting document management data
- sending, receiving and handling document data management messages
- making reports from the document management database, and
- acting as a platform for accessing the other programs used to edit
- the actual contents of documents.

## 4. MODELLING TOOLS USED IN THE RESEARCH

### 4.1 Choice of modelling methods

Two types of formal modelling tools are needed for the definition of the functionality of the ICPDM system, a method for modelling the tasks performed during the document management process, and a method for modelling the information carried by the chunks of information or objects describing the documents. The methods used in the definition of the international standard for the exchange of product data, STEP [ISO 1992] were chosen. For activity modelling SADT diagrams are used [Marca and McGowan 1987]. The conceptual information structures are modelled using the EXPRESS language as well as its graphical counterpart [CEN 1991].

### 4.2 Activity modelling - SADT

SADT diagrams are most often used to model physical work performed by humans and machines. In this project it has been used to model the functions of a software system. In SADT diagrams activities are denoted by square boxes (see figures 3 and 4). Each diagram usually contains between 3-6 activities. If a more detailed breakdown is needed this can be handled by blowing up one activity into a full diagram. In this way a large SADT model can consist of hierarchical diagrams on several levels of abstraction.

Activity boxes are linked by arrows. Arrows entering a box from the left show input of material or information. Often these arrows originate from another activity as outputs, but this is not mandatory. The functioning of activities is controlled by information, regulations etc. This is shown by control arrows pointing down to the activity. Activities are performed by mechanisms which are shown by arrows pointing up towards the activity. Typical mechanisms are persons, machines, software or on higher levels of abstraction organisations.

### 4.3 Conceptual information modelling - EXPRESS

EXPRESS is an alphanumerical data definition language which is used for defining conceptual data models. The central concept in EXPRESS is the entity. An entity can be viewed both on an abstract level (i.e. the point A) or by explicitly declaring its attributes (i.e. x, y and z co-ordinates). Each attribute has a name, which in general tells something about what the attribute represents, as well as a type which tells what type of data the attribute consists of. The data types of attributes can in addition to basic primitive data types also be other entities. When used as a data type of the attributes of other entities the internal data structure of an entity is hidden, and the detailed structure can only be found by consulting the entity declaration of the entity in question. This principle makes complicated schema much easier to read and also facilitates software development according to the principles of object-oriented programming.

```
ENTITY Storage_data;  
storage_type : SELECT (Paper_document, File);  
location : STRING;  
END ENTITY;
```

```
ENTITY File;  
file_name : STRING;  
program : STRING;  
file_format : STRING;  
backup_file : SET [0:1] OF File;  
END ENTITY;
```

```
ENTITY Paper_document;  
  
END ENTITY;
```

The attribute data types which are allowed in EXPRESS are:

- integer and real numbers, alphanumeric strings
- aggregated data typed such as lists and sets
- other entities
- functions
- enumeration of all allowable values.

In the graphical EXPRESS-G diagram format entities are represented by rectangles, with the name of the entity indicated inside the rectangle ( for examples see figures 5-7 in section 6). Attribute relationships between entities are represented by lines. In these the circle is attached to the entity which functions as the data type of the explicit attribute of the dominant entity. Aggregate data types in relationships may be indicated by abbreviations such as S, L followed by the cardinalities. Thick lines are used to symbolise supertype-subtype relationships. The subtype end of such a relationship is indicated by a small terminal circle on the line.

## 5. AN ACTIVITY MODEL OF THE TASKS PERFORMED BY THE ICPDM SYSTEM

The activity model consists of five main tasks (activities) shown in fig. 3 starting from receiving the document management information (box A1) and ending at distributing new document management information (box A5). The second and third activity are dealing with the documents themselves: to fetch or to create them (box A2) and to edit them (box A3). The fourth activity is to update the document management information (box A4).

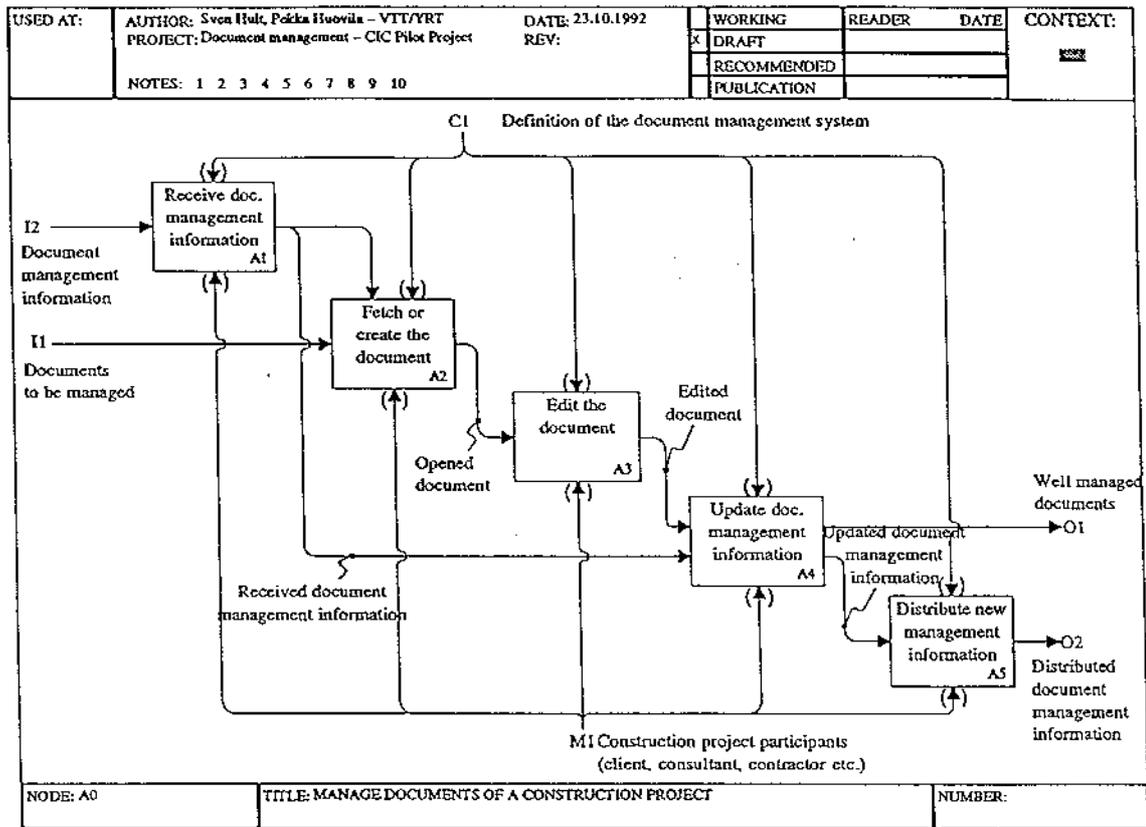


Figure 3. IDEF0 diagram A0.

All these tasks are controlled by the definitions of the ICPDM system and performed by the construction project participants. The first task gives an output of received document management information that controls the second task. After the document is fetched or created (second task) it is edited (third task). The fourth task is to update document management information. The outputs are well managed documents on the other hand and updated management information on the other hand. The latter is acting as input for distribution (fifth task).

Each of these main activities are then zoomed to be a lower level of abstract and presented as separate activity diagrams. As an example a more detailed breakdown of the first activity (A1 – receive document management information) is shown in fig. 4.

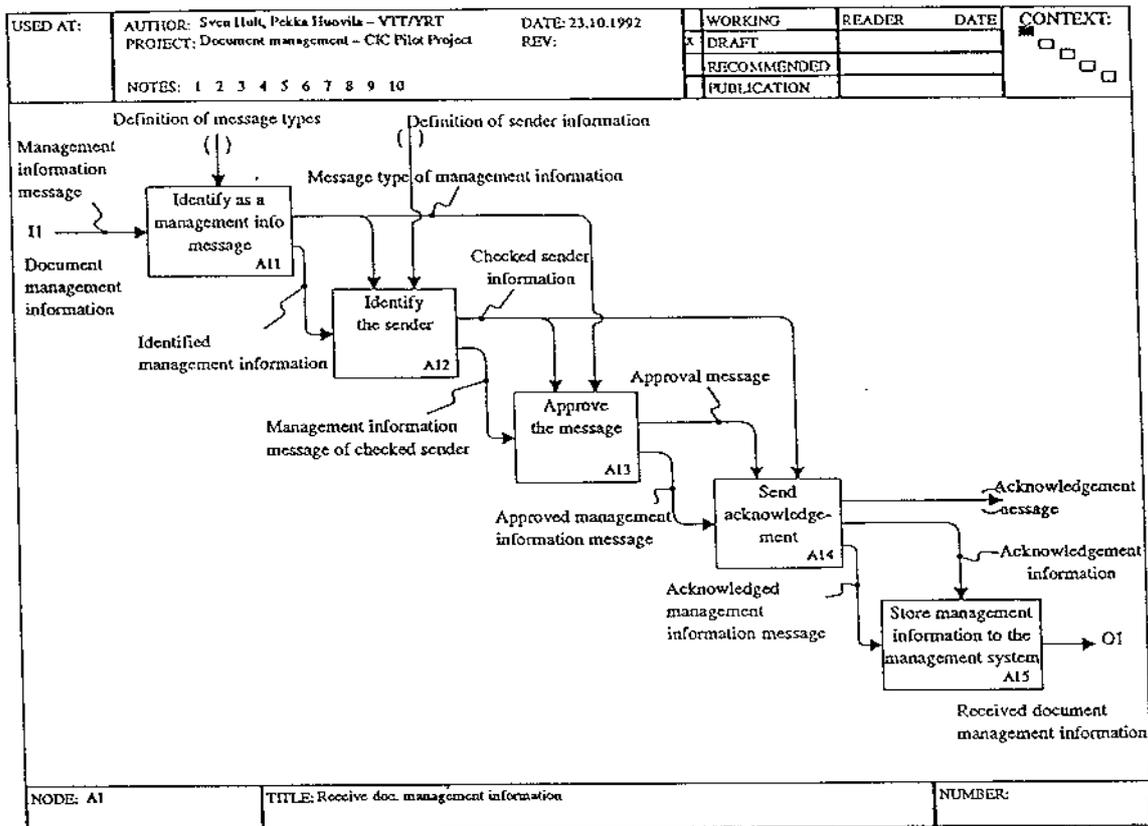


Figure 4. Receive document management information.

The diagrams have no direct connection to the timing of the activities – they only list the tasks and illustrate the relations between them.

## 6. A CONCEPTUAL MODEL OF THE INFORMATION MANAGED BY THE ICPDM SYSTEM

### 6.1 The document object

The conceptual model consists of three basic types of objects;

- Document object
- Project participant object

- Document distribution objects

The document is the main object of the ICPDM conceptual model. A document can be independent or it can belong to a construction project. It can consist of design data, project management data or contractual data. If a document is revised, the new version will be a totally new document. The new document will have a relation to the older version (the relation type is "previous\_version").

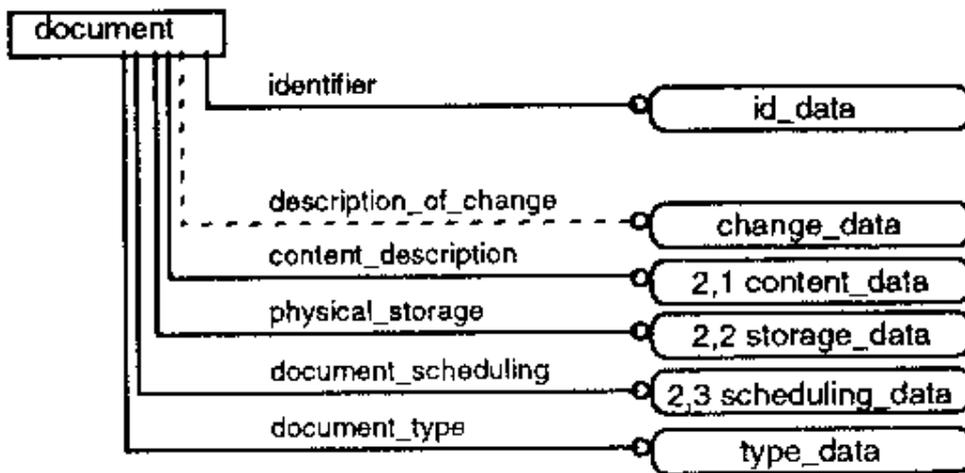


Figure 5. The document object and its main attributes.

A document has an identifier. If a document is a new version of a previously stored document, it has a freeformat textual description of changes made after the previous version. A document has a content description that can be textual or/and a scaled-down bitmap picture.

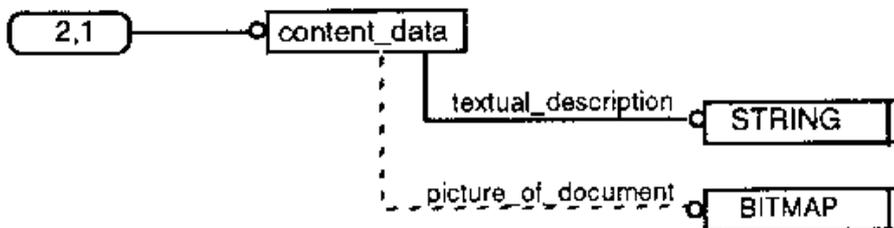


Figure 6. Content data.

The document has information concerning the way it is stored. If the document is stored electronically the object description includes the file name, the directory path, definition of

the program and the file format used when the document was created. If the document exists only on paper, the data of its storage can be inserted into the management base.

The document's scheduling data consists of planned and occurred completion dates, the document's state data (a document can for example have the following states: proposition, draft, accepted for construction, as-built etc.)

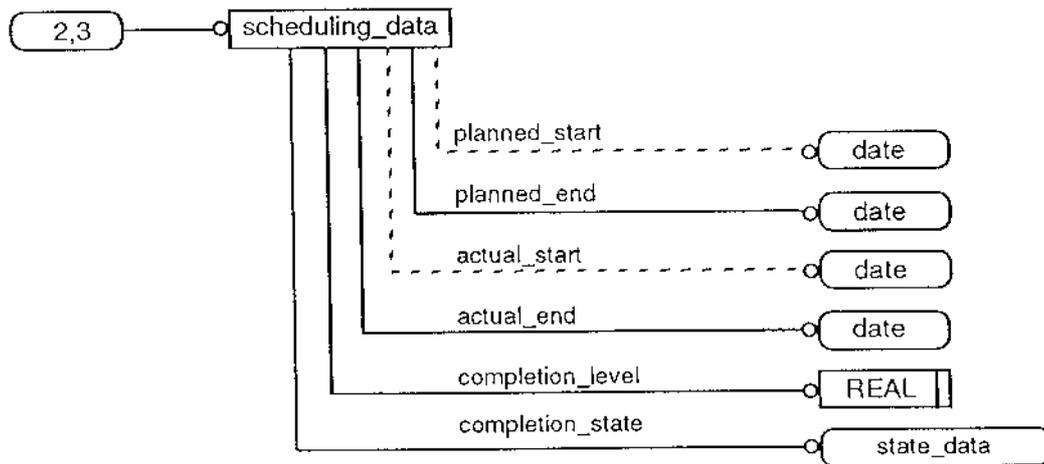


Fig. 7. Scheduling data.

A document has a type defining what kind of document it is. The three main types are

- o design document
- o project management document
- o contractual document.

The type attribute is a hierarchical list of types as show in figure 8. The first level defines the main type and next one or two levels define the document more specifically.

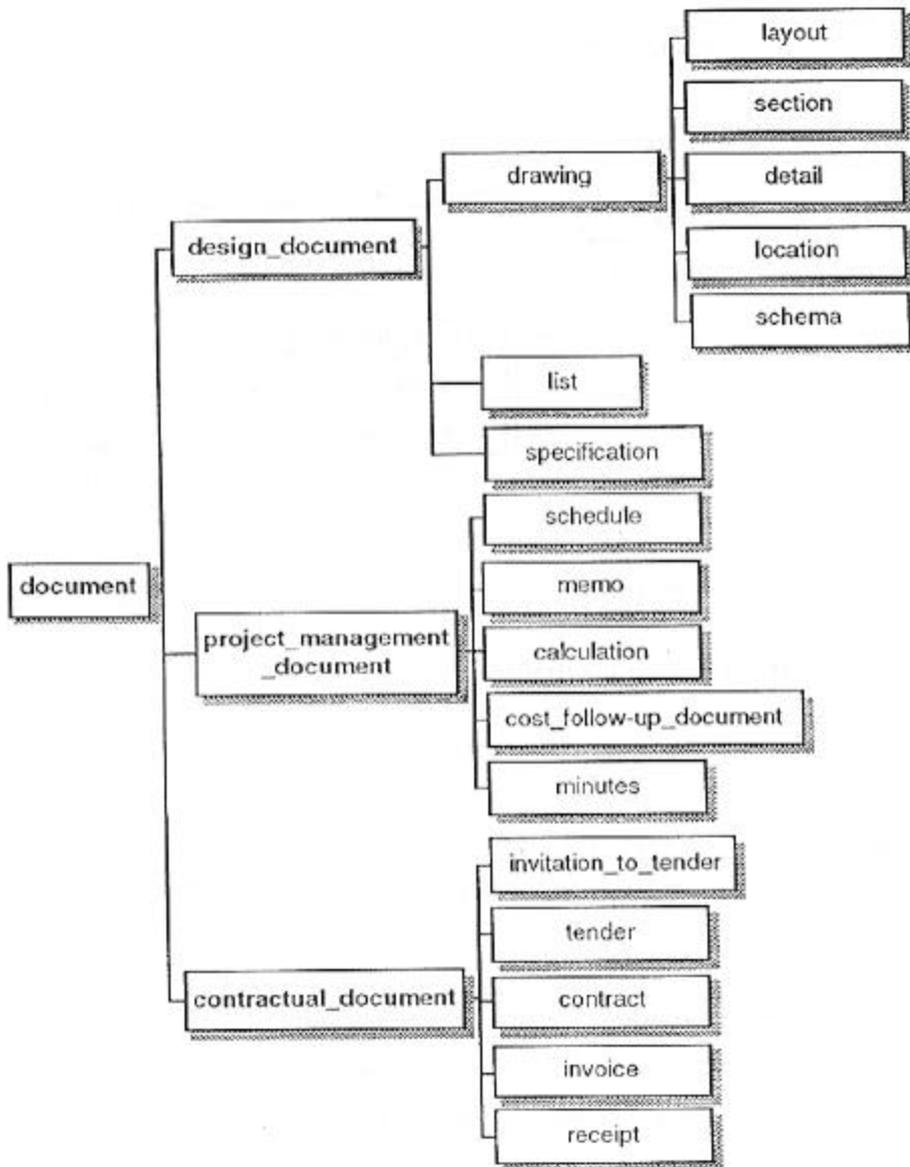


Figure 8. Document type tree.

Design documents have certain additional management information attributes lacking from the other disciplines. Design documents always have design discipline information that defines the scope of documents. The design discipline can be for example "Architect" or "HVAC Designer". This attribute allows queries using the design discipline as a search key.

One important subclass of design documents is a drawing. It has additional attributes concerning the scale of the drawing, and the artefact or subject described by the drawing. The user of the system can define which floors, sections or building parts are defined or shown in the drawing.

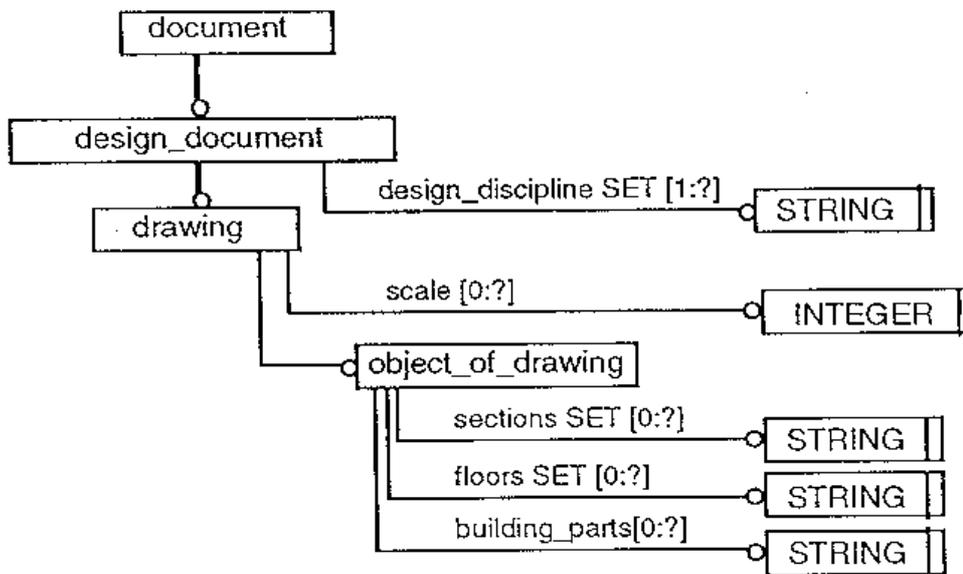


Figure 9. Document and its subclasses.

A document can have relations to other documents. There are three types of relations.

A `document_set` -relation defines a part-of type relation between an object describing a set of documents and its part documents. A collection document can be used to gather different documents to a specific use. A good example is provided by all the documents needed to define a contract's specifications, or all the documents needed to apply for a building permit.

A `source_document` relation defines which documents were used as sources when a new document was created. With this relation one can find all documents that need some sort of inspection if the source document is changed.

A `previous_version` relation shows which document was the previous version of a document.

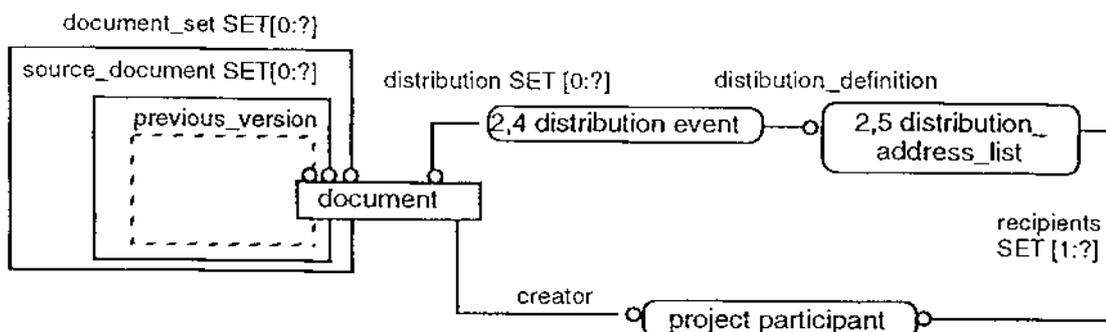


Figure 10. A document's relations to other objects.

## 6.2 Project participant object

A document has a creator. The creator is defined by a relation to the object "project participant". A project participant can be an organisation or a person working in an organisation.

## 6.3 Document distribution objects

When a document management data is sent to an other party, the system creates a distribution event and a relation "distribution" between it and the corresponding document. A distribution event is an object that defines one specific distribution event in which a document is sent from its creator to another party. A distribution event has an id, time of the event and recipient data as its attributes.

Recipient data is actually a relation to a "Distribution address list" object. A distribution address list has an identifier, and items of a list. Each item has a relation to a project participant object and user access rights data. User access right defines how the recipient can access the actual data of the document. There are three different values; read\_and\_write, read\_only and exists.

## 7. CONCLUSIONS

The research described in this paper draws on earlier work concerning overall strategies for CIC and product modelling [Björk 1992a]. The authors believe that it points out a new and slightly differing approach to integration of the total information produced during a construction project. Consequently it complements the contributions of other authors who have focused more on construction process modelling [Sanvido 1992], construction information classification systems [Karlsson and Allott 1991] or conceptual modelling [Froese 1992].

One aspect of VTT's current project which we believe is innovative is the application of formal modelling methods from the product modelling domain to this particular subject area. This is important in view of a future integration of ICPDM systems and product model based systems.

The hypothesis underlying this effort is that this type of ICPDM systems could enhance the overall management of the information generated during a construction project, and that this enhancement will lead to increased efficiency and better quality through a decrease in errors. This type of systems would obviously also be beneficial as integrated parts of quality systems.

## ACKNOWLEDGEMENTS

The research described in the paper is based on ideas developed by Bo-Christer Björk during earlier projects at VTT focusing on product modelling and on defining an overall framework for construction computing. Pekka Huovila manages the ICPDM project. Sven Hult has co-ordinated the definition of the activity and conceptual models and is responsible for the software implementation of the ICPDM. The research is funded by the Technology Development Centre of Finland.

## REFERENCES

1. B.-C. Björk , A unified approach for modelling construction information, *Building and Environment*, Vol. 27, No. 2, April, pp. 173-194., (1992).
2. B.-C. Björk, Information versus document level data transfer in computer integrated construction, preproceedings of the international workshop on Frameworks for Computer Integrated Construction, VTT, Espoo, Finland, 5-9.10. 1992.
3. CEN, Express Language Reference Manual, CEN/CLC/AMT/WG STEP N48, Association française de normalisation, (1991) Paris, 160 p.
4. A. Dupagne edt., Computer Integrated Building, Strategic Final Report, Esprit II: Exploratory Action No 5604, CE Commission D. G. XIII, December 1991.
5. T. Froese, Integrated Computer-Aided Project Management Through Standard Object-Oriented Models, Doctoral dissertation, Dept. of Civil Engineering, Stanford University, June (1992).
6. H. Karlsson, and T. Allott, Classification of Information in the Construction Process Draft ISO Standard, ISO/TC59/SC13, (1990).
7. ISO ,The STEP standard, Future ISO standard 10303. Available in several continuously developing parts through electronic mail via the National Institute for Standards and Technology, Washington D.C.,email address Internet nptserver@cme.nist.gov, (1992).
8. D. A. Marca and C. L. McGowan, SADT - Structured Analysis and Design Technique McGraw-Hill, New York, (1987).
9. J. Neutebooms, Bridging the Gap Between EDI and PDI, Preproceedings of the seminar Technical Data Interchange in Construction, CSTB, Paris, 1-2.10.1992.
10. V. E. Sanvido, Linking levels of abstraction of a building design *Building and Environment*, Vol. 27, No. 2, pp. 195-208, (1992).

## APPENDIX ICPDM SCHEMA VERSION 1.0

SCHEMA ICPDM-model;

```
ENTITY Document;
SUPERTYPE OF (Design_document);
    identifier : STRING;
    description_of_change : OPTIONAL STRING;
    content_description : Content_data;
    physical_storage : Storage_data;
    document_scheduling : Scheduling_data;
    document_type : Type_data;
    creator : Project_participant;
    distribution : SET [0:?] OF Distribution_event;
    previous_version : OPTIONAL Document;
    source_document : OPTIONAL SET [0:?] OF Document;
    document_set : OPTIONAL SET [0:?] OF Document;
    UNIQUE
    identifier;
END ENTITY;

ENTITY Design_document;
SUBTYPE OF (Document);
SUPERTYPE OF (Drawing);
    design_discipline : STRING;
END ENTITY;

ENTITY Drawing;
SUBTYPE OF (Design_document);
    scale : INTEGER;
    sections : OPTIONAL SET [0:?] OF STRING;
    floors : OPTIONAL SET [0:?] OF STRING;
    building_parts : OPTIONAL SET [0:?] OF STRING;
END ENTITY;

ENTITY Content_data;
    textual_description : STRING;
    picture_of_document : OPTIONAL BITMAP;
END ENTITY;

ENTITY Storage_data;
storage_type : SELECT (Paper_document, File);
location : STRING;
END ENTITY;

ENTITY Scheduling_data;
planned_start : OPTIONAL DATE;
planned_end : DATE;
actual_start : DATE;
actual_end : DATE;
completion_level : REAL;
completion_state : SELECT ("draft", "proposition", "inspected", "ready");
```

END ENTITY;

ENTITY File;

file\_name : STRING;  
program : STRING;  
file\_format : STRING;  
backup\_file : SET [0:1] OF File;

END ENTITY;

ENTITY Paper\_document;

END ENTITY;

ENTITY Distribution\_event;

event\_id : STRING;  
distribution\_time : Date;  
recipient : Project\_participant;  
distribution\_criteria : Distribution\_address\_list;

INVERSE

distributed\_document : document FOR distribution;

UNIQUE

event\_id;

END ENTITY;

ENTITY Distribution\_address\_list;

list\_id : STRING;  
distribution\_list\_item : SET [1:?] OF List\_item\_data;

INVERSE

user\_event : SET [0:?] OF Distribution\_event FOR distribution\_criteria ;

UNIQUE

list\_id;

END ENTITY;

ENTITY List\_item\_data;

user\_access\_right : STRING;  
recipient : Project\_participant;

END ENTITY;

ENTITY Project\_participant;

participant\_id : STRING;  
function\_in\_the\_project : STRING;  
address : STRING;  
works\_for : Project\_participant;

INVERSE

address\_data : SET [0:?] OF List\_item\_data FOR recipient ;

document\_created\_by : SET [0:?] OF Document FOR creator;

UNIQUE

participant\_id;

END ENTITY;

END\_SCHEMA.