THREE PROTOTYPE CDDM SYSTEMS

Žiga Turk, PhD, Assoc.Prof.
University of Ljubljana, Department of Civil Engineering,
Jamova 2, Ljubljana, Slovenia; email: ziga.turk@fagg.uni-lj.si

Abstract

Computer Integrated Construction (CIC) is a kind of civil engineering production where by using computers and information technology, high degrees of integration and data transfer are achieved. On the long run software supporting CIC relays on the definition of the conceptual building product and building process models. These models are large because a lot of various data structures are needed to describe them; and they are complex, because the data are related to each other in many ways. Functional solutions for CIC may also be achieved by diminishing the level of detail at which the information is observed. The larger data chunk to an object or an attribute is a document which may contain (or refer to) many objects. Construction document management (CDM) systems are a shorter term solution for CIC; they are not as numerous and they hide complexity of the information they carry. In the life cycle of a building the information creation, demand and use are greatest in the pre-construction phases. In the paper three approaches to the construction design documentation management (CDDM) are described. The process based one is presented in most detail.

1 INTRODUCTION

The section will introduce integrated CAD and the role of CDDM systems (CDDMS) in the CIC effort.

1.1 ON COLLABORATED AND INTEGRATED CAD

The complexity of engineering artefacts requires the employment of the most typical tool for fighting complexity - decomposition. On one hand artefacts are decomposed into components or assemblies which are then designed separately; on the other, the design process of a complex component is broken into many discrete design steps. Each is then designed separately in a design sub-process. These sub-processes may run concurrently or sequentially by one or more designers. When the complexity of the design increases, the number of sub-designs or sub-processes grows. Since they are not entirely independent, the communication between the sub-processes is very important. In the past, the communication media was spoken word for synchronous communications i.e. meetings of design teams and written words and drafts for asynchronous communication i.e. between the designers and the manufacturers.

Lately a growing number of design-sub-processes are supported by information technology (IT) and the result is information that is written digitally. The motives for communicating this information in the same - digital - format are:

- it makes the information readable by programs that are used in other design-sub-processes;
- it ensures that there are no distortions or information loss by copying;
- it preserves the accuracy of information because no errors can be introduced through human interpretation of the information or by retyping the information ;
- it is compact and requires less (physical space) than information on paper;
- it is up to date and consistent; digital format stored in the computer can be updated as the project advances.

The kind of design where information technology is the main integrative element between the design sub-processes and their results is called *integrated computer aided design* (ICAD). A kind of design where IT is the main integrative element between the participants of the design process (designers) is called *collaborated* CAD.

1.2 PRODUCT, PROCESS AND DOCUMENTATION MODELS

Both technologies can be applied on different scales of decomposition of the designed artefact. The difference is in the size of the unit of information that is handled by software that supports integrated or collaborated CAD. On the small scale a unit may be as small as one single attribute like a thickness of a wall. On a larger scale a unit may be a document or a set of documents that are a result of a design sub-process (like dimensions of all concrete elements in a building).

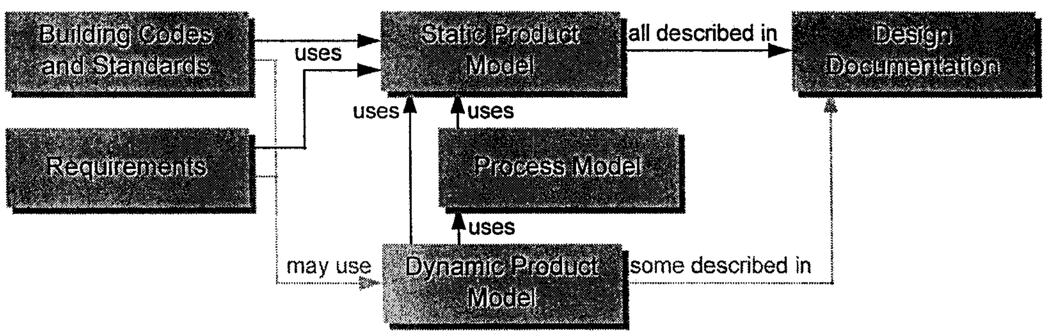

Fig. 1: ICAD modelling spaces.

In (Turk, 1991) the ICAD was decomposed to five distinct modelling spaces as depicted in Fig. 1. The client-server relations reveal the principal means of interaction between them. Fig. 2 shows how are these relations manifested on a small fraction of a product model. It also demonstrates one of the main obstacles in attaching the process related information. It seems impractical to attach process related information like the one suggested in IRMA (Luiten et al, 1993) model to just about every attribute in the product model. To find larger units for the attachment of process related information is one of the most exciting tasks of the present CIC research. It may take quite some time before the product and process models are defined and combined into a unified model.

In the meantime we propose a document as a unit onto which the process related information should be attached and a document management system as a replacement for a full product model management system. The three prototypes discussed in this paper manage the documents by adding additional meta information to them.

2 PARTIAL PROTOTYPES

The section will summarise the features of two older CDDMS prototypes in which the first stressed the automatic design paradigm and the second placed the control into the hands of the designer.

2.1 FRAME BASED PROTOTYPE

The frame based prototype was built in 1991/92 (Turk, 1992) and modelled the design environment using the model-view-controller paradigm as views, controllers, processes and managers. The prototype used expert system (ES) shell Kappa (Intellicorp) and it's frame based technology to model these entities.

The views are hierarchically classified according to the information they contain. There are two subtypes - the "true" views and the files. The firsts are interfaces to databases describing a product (usually a collection of files and related manipulation tools). The controllers are abstractions of the existing programs and are not hierarchically ordered. A process is an abstraction of a part of the design process and connects a view to a controller. Managers are humans or special programs.

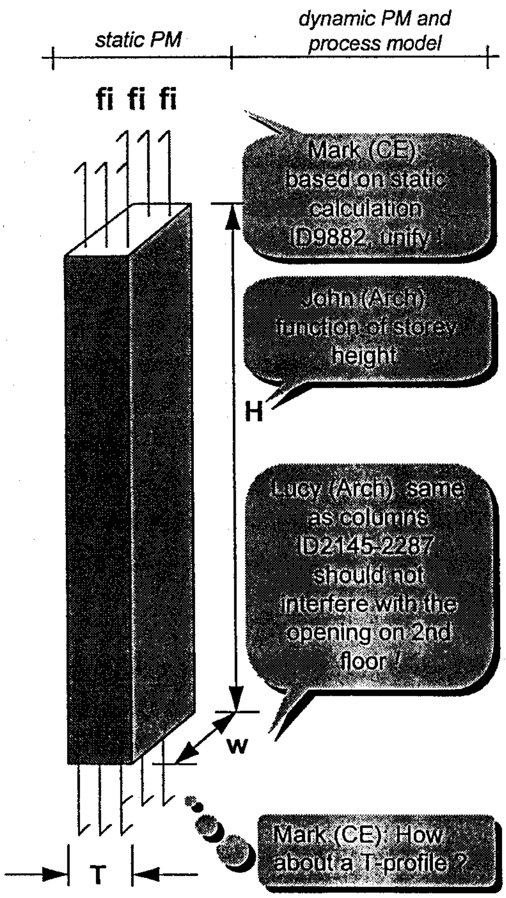

Fig. 2: Static product model and (in columns) dynamic product model and process model related data about a column.

The design process is understood as a goal satisfaction problem: the goal of a design process is to create a certain document and, at all times, ensure that it is up to date with regard to other documents on which this document is based. Expert system's chaining mechanism and goal oriented reasoning is used to do that. If a document would become obsolete, an expert system would start a process, that is, run a controller on a particular document, so that it would be up to date again. The environment has attributes of a blackboard system where various controllers assert changes of the information they control to the goal

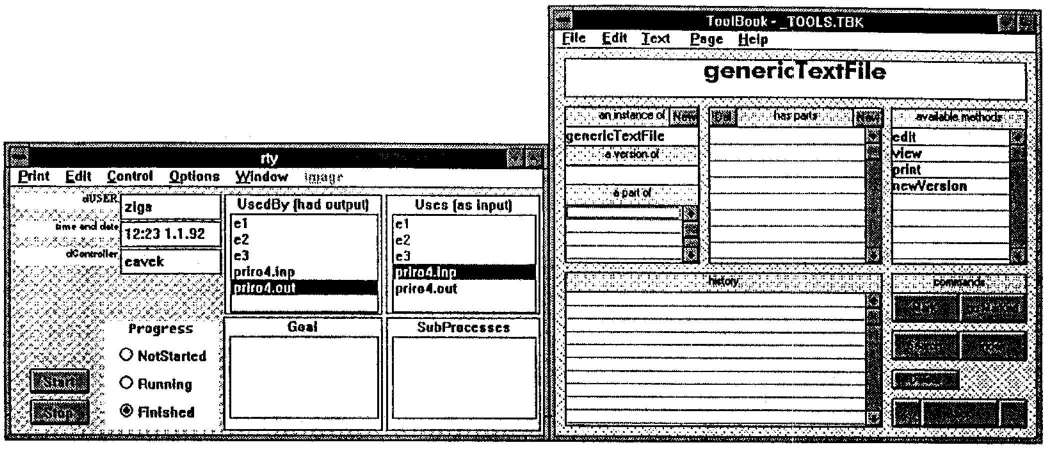

Fig. 3: Forms of the frame based (left) and the hypermedia based prototypes.

stack so that other controllers can respond to it. The reasoning mechanism has the function of the process dispatcher. A typical form showing information about a process is depicted in Fig. 3 (left).

The prototype tackled some interesting issues to design process management like the automatic supervision of the design process and explored the usefulness of ES shells for CDDM systems. It was well suited for the management of information that resulted from non-interactive computer programs. It was less efficient on problems and software where human interaction was required - and such is the majority of design related software.

2.2 HYPERMEDIA BASED PROTOTYPE

Hypermedia based prototype takes the opposite approach and attempts to support the design process by creating a user friendly environment where the human designer could be making the decisions while the software would be keeping track of the data. It has no ambition to control the design process in any way but to provide easy access to information needed for that. It was built upon a hypercard like product Toolbook (Asymetrix) and uses cards as an abstraction for a collection of construction information. A card represents a file or a set of files. Since Toolbook and other cardware software enable typing of page backgrounds these are typed according to the tools that may be used to manage the information they contain. The backgrounds are hierarchically ordered.

The information model behind this prototype is simpler then the one used in the frame based prototype. The user is not manipulating processes but documents. Each document has an associated meta-information which is manipulated through a form such as the one shown in Fig. 3-right. It has a type and associated methods, version control information, part-of and has parts associations and a free form history list. The form's methods may be run directly from the form and would manipulate the contents of the document.

3 PROCESS BASED PROTOTYPE

Based on the experience from the previous two prototypes the primary role of IT was decided to be that of a medium - a medium which is (1) connecting the design sub-processes, the computer based design tools with the designers - and (2) connecting the designers themselves - creating the environment for a true collaborative design.

3.1 GOALS

The goals for developing the process based version were to:

a) Improve the deficiencies of the older prototypes with regard robustness of storage; the relation between the documents and the meta data was very volatile and thus not well accepted by the users;
b) Support collaborative design by ensuring central and transparent storage that may be accessed through local and wide area networks; both previous prototypes were limited in tools and in concept to a single workstation. Networking is essential for collaboration intense tasks such as design;
c) Support features of a quality controlled system; The solution should provide information onto which systems for quality assurance as suggested in the ISO 9000 series could be attached;
d) Provide a base for design scheduling and management using IT;
e) Support document versioning and help assure design consistency.

3.2 MODEL OF THE DESIGN ENVIRONMENT

This section provides "plain English" description of the design environment and the design process. In part it is influenced by the work of Bjork et al, 1993. The process prototype is based on a CDDM system model where the key entities are:

- *design task* or sub-task which results in a
- *document* which is created or supervised by a
- *designer* using a
- *tool.*

These entities are described in greater detail below.

3.2.1 Design task

Design task is part of a design process. It:

- belongs to a design process;
- may be part of another design task so that infinite structuring of the design process is possible; it can not be (at present) part of many design tasks.
- has progression state which may be one of: planned, started, suspended and finished;
- has progression value which is a percentage showing how much of a task has been estimated to be accomplished;
- has planned and actual start and finish;
- has a supervisor, who manages this task;

- has free form description and comments of arbitrary length.

3.2.2 Design document

Design document is a result of a design task. It has its contents and an envelope. The contents of a document may be a computer file or some not computerised media and it is entirely independent from its envelope. Envelope knows about the type of the content but is not concerned with any of its details. Other features of a document are:
- belongs to a document type (listed below in section 3.2.3);
- is result of one design task;
- has author and supervisor;
- has progression status which is one of: empty(nothing done yet), draft (being worked on), proposal (author is done with it), checked (document is cleared by the supervisor).
- has a physical location where this document is stored;
- may be discussed or commented;
- has configuration information. A configuration is a set of documents that together define a design alternative. Configuration "main" means that the document is part of the main version of the design. Any other value suggests that the document is part of an alternative solution.
- may have base documents; these are documents which contain information that that was used to create this document;
- may have previous versions on which are *replaced by* this document or to which this document is an *alternative to*. In the last case the document also has configuration information other than "main".

3.2.3 Types of documents by storage

Documents are typed according to the location of their physical storage:
a) documents not stored digitally;
b) documents that are physically stored in the envelope (within Windows environment documents made with applications that support OLE);
c) document stored in files of the network file system;
d) document stored on remote file system and need to be copied to the local file system, before their content can be examined.

Documents of type c and d are further sub-typed according to the file type.

3.2.4 File types and file tools

Files are types and the associated tools are used to create an object oriented interface to a particular file. File type contains information about:
- file type name;
- name of mostly one the super file type. A file type from which files of this type inherit tools.
- a list of tools that may be applied to instances of this file type.

File tools are pieces of code (typically programs) that may be executed with files of an associated type as a parameter. File tool contains information which is needed to run the program:

- name of the tool;
- the associated type;
- command that runs the tool;
- parameters of the command;
- implementation details of the command.

3.2.5 People and companies

People involved in the design project are managers of tasks, supervisors and authors of documents. Information associated with them is:

- personal information like name, surname, date of birth, picture ...
- professional information (title, education);
- organisational information (position within the company);
- employment information - company they work for;
- mailing and electronic address.

Companies employ people. The associated information are name, address and main contact person.

- name;
- address;
- contact.

3.2.6 Collaboration support

With respect to the project, designers communicate with each other in two ways:

- one to one - the communication is private;
- one to many - designers comment documents, configurations and tasks or other comments.

3.3 IMPLEMENTATION

The section describes the implementation of the above schema in a relational database (RDB), the environment and the user interface.

3.3.1 Database schema

The schema is depicted in Fig. 4.

3.3.2 Implementation environment

The prototype is running under Microsoft Windows 3.1 and relational database system Microsoft Access. It is written in Access macro language and Access basic. Minimum hardware configuration is a 386 with 4MB of RAM. Runtime or full version of MS Access is also needed.

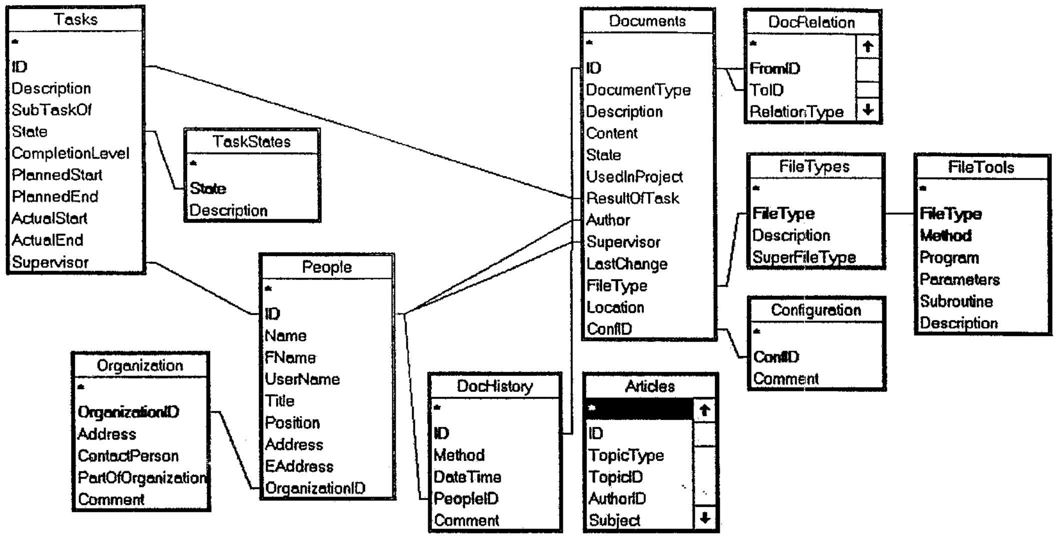

Figure 4: Database schema. Tables are shown as boxes, fields as words in those boxes and relations between tables as lines.

It supports any network operation system (NOS) that has the functionality of a network file system (NFS). This is a system where remote disks are logically presented to the applications as if they were local. Examples of such systems are Novell Netware, Windows for Workgroups, Lantastic and PC/NFS. These systems usually operate on a local area network (LAN) which is usually limited to a local building or buildings (Fig. 5). Using the IP tunnelling, however, some of the NOS make it possible to access information on media anywhere on the wide area network (WAN) as if it were on the local hard disk, usually at a much lower speed. Trends in the WAN speeds and the introduction of technologies like FDDI or ATM make it probable that in the near future access to distant files on the WAN within an industrial area will be as fast as today on a LAN.

The implementation also provides alternative means to access information residing on machines attached to WAN. These can be virtually anywhere in the world. The program builds on the hypertext transfer protocol (HTTP) and the World Wide Web (WWW). The environment uses a free client program which can display information on remote machines or make a copy of the information on the local machine (Fig. 6). It is only possible to view but not to change the remote information. The information accessed through the WWW is expected to be of reference nature and not part of one project only.

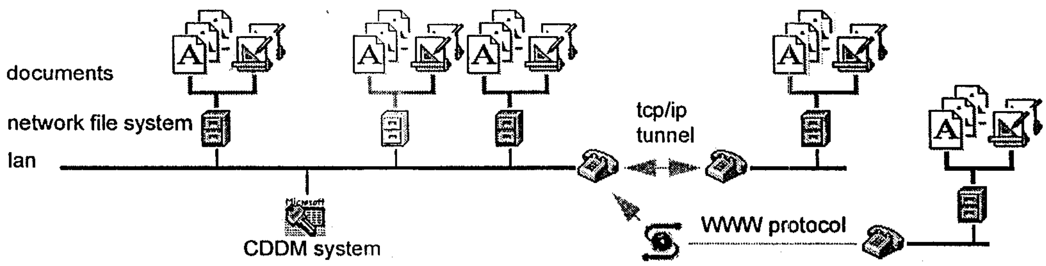

Fig. 5: Networking environment and the three means to access documents.

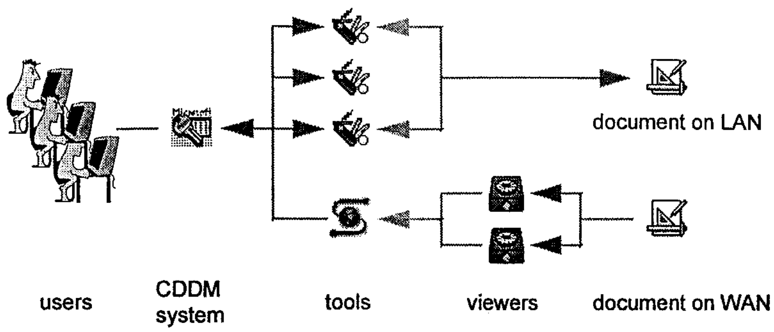

Fig. 6: Accessing documents - user's perspective.

Communication is supported with an interface to an external program for sending and reading electronic mail. Sending e-mail to authors and supervisors of documents is particularly easy - both the address and the subject field are filled in by the system with designer's and document's ID.

One to many communication is supported through a simple but effective conferencing subsystem that functions as part of the environment. It is thread based. A thread is a set of articles that were created as direct or indirect replies to one single "root" article. In this case every task, document or configuration has the role of a "root" article and may be commented and discussed by other designers. A more formal approach to discussions and decision making like IBIS (Kunz and Rittel, 1970) is being considered.

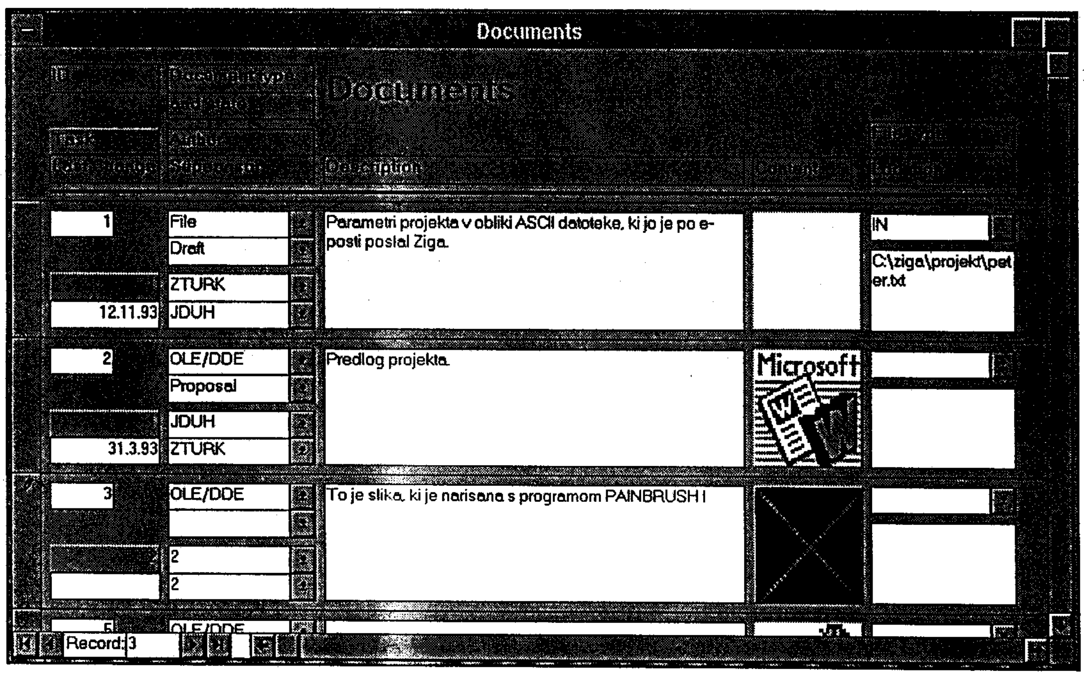

Figure 7: Document form shows multiple documents at once since it is expected, that it will be the most used form. The content field either shows the icon of the program which is used to manipulate the document or is empty when the document is not a result of an OLE supporting application. In the first case

double clicking on the icon opens the application, in the second it opens up a list which lists the programs that may be used to manipulate the document.

3.3.3 User interface

The program is form oriented. User interface employes some of the gadgets of the Windows operating system like fields, menus, buttons, drop down menus, and scroll bars. It is point-and-click rather than drag and drop. Some sample forms are in Fig. 7 and 8.

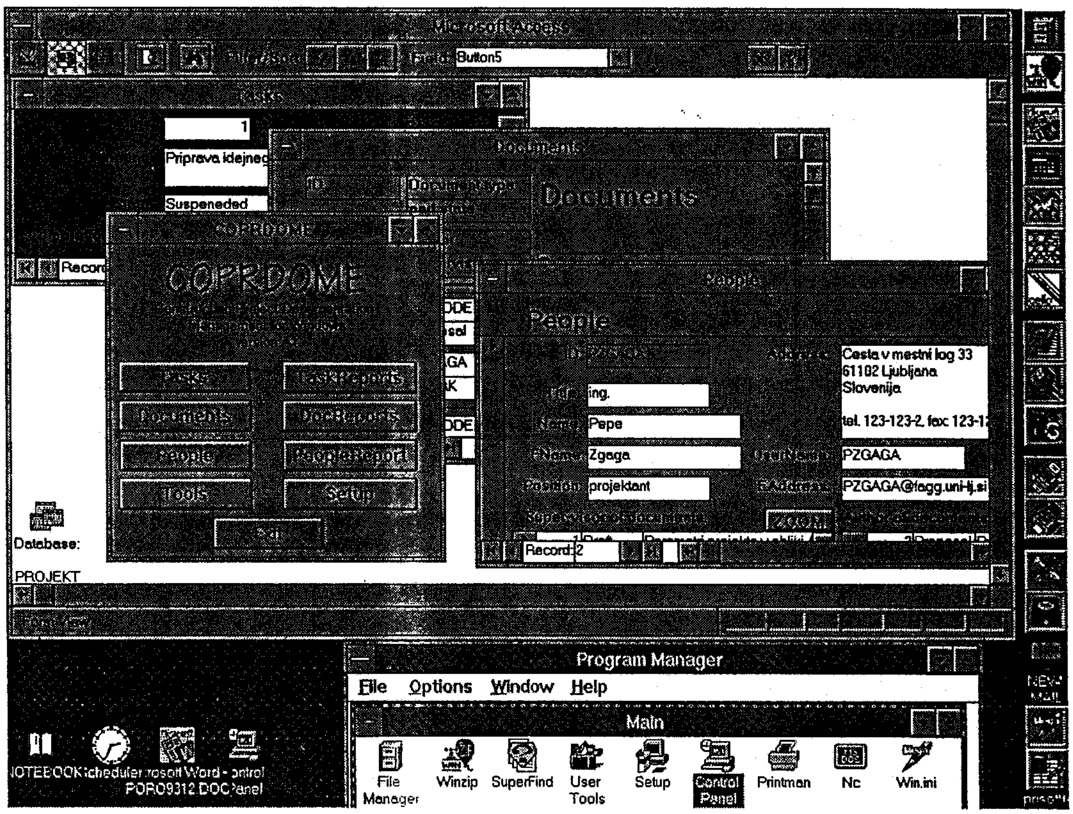

Figure 8: Windows screen when using the program.

4 CONCLUSIONS

A prototype of a construction project documentation management system was developed. Most goals listed in section 3.1 were achieved. This section will point out the main deficiencies of the presented version and outline further work.

The program competes with some functions of advanced file systems (attaching additional information to files, handling remote files ...), of operating systems shells (running programs), scheduling and group-ware programs. The present version can not compete with any of those but it can, however, compete as a combination of the four which is specially adapted to engineering needs. OLE servers solve, in part, the main technical problem of such environments - the relation between the document envelope and the actual document. The OLE documents are stored in the actual database with their envelope. Files may still be handled not only through the environment but directly through the operating system. Windows environment provides no means to protect or restrict the interface to those files and it is quite possible that the user deletes a file that is referenced by the envelope without telling the envelope about it. A possible solution to this problem would be a use of a virtual OLE server that would create an OLE interface to non OLE applications. The alternative would be the use of an object oriented operating system like NextStep or OS/2.

The process based prototype also showed that by using Windows based relational database management tools, OLE, industry standard local and wide-area network tools, the development of core functions of such environments is quite fast. On the other hand many functions of the environment remained raw. For searching and reports, the user is expected to be able to use the corresponding generic operations that are part of the MS Access. For some of the most popular queries and searchers special, user friendlier ones would be welcomed. Current prototype's support for document versioning and dependencies between documents is limited. It keeps track of the relations between documents but leaves the management responsibilities to the users.

Further development work will be focused towards evolving the prototype into an end-user version to be distributed. Further research work is focused on improving version and dependencies management, the extension of the scope from the design phase to other life cycle stages and to establishing a clear relation with the product models. Paper (Turk et al, 1994) is a step in that direction.

5 REFERENCES

Björk B-C., Huovila P., Hult, S. (1993). Integrated Construction Project Document Management, Paper submitted to the EuropIA'93 conference.

Kunz W. and Rittel H. (1970). Issues as Elements of Information Systems. Report S-78-2, Institut für Grundlagen der Planung, Universität Stuttgart, Stuttgart, Germany.

Luiten G., Froese T., Björk B-C., Cooper G., Junge R., Karstila K., Oxman R. (1993). An information reference model for architecture, engineering and construction, in Mathur K.S., Betts M.P., Tham K.W. (eds) Management of information technology for construction, World Scientific Publishing, Singapore.

Turk Ž. (1991). Integration of Existing Programs Using Frames, CIB Seminar Computer Integrated Future, 16-17 September, Eindhoven, Netherlands.

Turk Ž. (1992). Environment for the computer integrated design of structures, PhD thesis, University of Ljubljana, Ljubljana, Slovenia.

Turk Ž., Björk B-C, Johansson K., Svensson K. (1994). Document management systems as an essential step towards CIC, accepted paper; CIB W78 Workshop on Computer Integrated Construction, VTT, Helsinki, Finland, August 22-24, 1994.