# A METHODOLOGY FOR DYNAMICALLY ASSEMBLING AND MODIFYING AN ACTIVE-REPOSITORY BUILDING DATABASE

Ahmed Mokhtar[1], Claude Bédard[2], Paul Fazio[3]

## ABSTRACT

An information model has been developed at the Centre for Building Studies, Concordia University to facilitate the management of design changes through inter-disciplinary coordination. The main component in the information model is a central project database that can be characterized as an active repository of building components data. The data in this active repository communicate design changes among affected disciplines and help ensure compatibility of design information. Because buildings are generally unique, a fixed data structure for building components in such an active repository is impractical. This paper proposes a methodology that enables managers of multi-disciplinary design teams to dynamically and transparently assemble and modify the data structure for the active repository of building components in such a way as to suit a variety of specific building design configurations. The methodology relies on combining some of the capabilities of the object-oriented technology with the flexibility of the relational database technology. To enable the implementation of such a combination, the project database is divided into building components data part and a management data part. The management data allows the model to take advantage of SQL capabilities in controlling the data structure of building components. Design managers are also given the option to assemble the data structure from reusable elements that contain both the building components attributes and the rules that are necessary for managing design changes. These reusable elements are gathered into three levels of increasing complexity: components, systems, and configurations. Such organization greatly facilitates the structuring of the project database and increases the reusability of elements. The paper details the methodology and demonstrates its use with a simple example.

## INTRODUCTION

Buildings are unique products. A hospital contains very different components from those of a warehouse. Furthermore, a warehouse may share some components with other warehouses but not all. For example, an air-heated warehouse may contain air distribution ducts while a radiant heated warehouse will not. The definition of a data structure that carries the design information of building projects may therefore follow either one of the following two strategies. The *first* strategy is to create a universal data structure that covers all the possible building projects. The *second* strategy is to customize the data structure to suit every individual building project. The *first* strategy should result in a reusable and

[1] Ph.D. Candidate. [2] Associate Professor. [3] Professor and Director.
Centre for Building Studies, Concordia University. 1455 de Maisonneuve Blvd. W, Montreal, Quebec, Canada H3G 1M8

therefore desirable data structure, yet it requires the inclusion of a complete set of available and newly introduced building components. As the number of these components is virtually unlimited, this strategy rapidly becomes impractical.

At the Centre for Building Studies, the *second* strategy is being explored. Such an exploration is part of the development of an information model that aims to facilitate the management of design changes through inter-disciplinary coordination during the detailed design stage. The information model utilizes a central project database that carries an active repository of building component data. When a design change is specified by a designer in a certain discipline, this data become active and notify all other affected design disciplines with the change. That is achieved through rules that supply the building data with the linking knowledge that is necessary to achieve the task.

The focus of this paper is to propose a methodology for customizing the structure of a central project database through the dynamic assembly and modification of active components. The paper starts by reviewing related research efforts. It presents next the proposed information model. Then the methodology is introduced in detail and is followed by an illustration of its implementation.

## REVIEW OF RELATED WORK

Modeling a database structure that carries building design information of is the focus of many research efforts in various parts of the world. The AEC Building Systems Model in the USA (Turner 1990) adopted a hierarchy of four levels which are system, system component, system component port, system component port joint. The RATAS model in Finland (Björk 1989) distinguishes five functional levels for building components. They are building, system, subsystem, part and detail. The COMBINE IDM model in Europe (Dubois and Parand 1993) has its root object as the Construction Project which may contain one or more Buildings. The Building is seen as an assembly of systems which are: Spatial System, Fabric System, Technical System, Functional System and External Environment System. Another classification has also been given in the Construction Project Reference Model (Rezgui and Depras 1995) where the building is described according to five complementary systems: structure, work, space, technical and separation systems. The purpose of this model is to enable the generation of construction documents from a single source of information. A multi-dimensional model of buildings (3P Model) which integrates the process, product, and participant is developed by Bédard (Bédard and Rivard 1995). It decomposes buildings as a product into structure, envelope, services and interior systems. Other research work has been conducted to create an environment and a set of concepts that link building models to various aspects of the building delivery process in areas like early design cost control (Tsou 1992), computer-aided architectural design (Turner 1992), conformance with regulation (De Waard 1992), evaluating building performance (Augenbroe 1992), maintenance information management (Svensson 1993), design and construction management (Sadri 1993), design of precast concrete facades (Karhn 1997). Another notable effort in using database to record building design data is through the International Alliance of Interoperability (IAI). The IAI is an alliance of the building industry that aims to integrate the AEC/FM industry by specifying Industry Foundation Classes (IFC) as a universal language to improve the communication, productivity, delivery time, cost, and quality throughout the design, construction, operation and maintenance life cycle of buildings (IAI 1996). IFC is a library of commonly defined objects that create “intelligent” project data such as the properties, behavior and graphical representation of building components (Herald 1997).

Exchanging design information among designers has been investigated by Vries and Somers (1995) who developed a process model for that purpose. They suggested that a protocol to exchange information should ensure that both senders and receivers of information should interpret it the same way. Also all the required and correct information should be exchanged. Abeu-zeid and others (1995) proposed shared-data resources as a primary component for common communication channels to exchange data between participants in a project. They used the structured analysis technique to define data shared between participants.

Management of design information was also the focus of some research work. Platt (1996) focused on design management of civil engineering projects through process modeling rather than data centric modeling. The process modeling has information on how a product is transformed from its initial state to the final deliverables. Rezgui has developed an approach to tackle the problem of integrity and consistency in construction documents production and management (Rezgui and Depras 1995). It is based on building a construction project reference model and a document reference model. Both models are linked with an association model that indexes building components to documentary items. A data model has been developed by Eastman et al. (1995) to support integrity management. The model which is called EDM-2 includes constraints that check the status of rules and specific structures that represents the knowledge embedded in an application.

The issue of managing design changes has not received much attention in the literature. One notable exception is a project by Krishnamurthy and Law (1995) which proposes a three layered model of versions, assemblies, and configuration. The model monitors independent design activities by systematically tracking components descriptions in individual disciplines. It achieves coordination through asynchronous communication of design changes.

In summary, the building industry is still in the exploration stage on how to support design information in a structured database. No standard is currently available to follow. In addition, few research work is available on design management, exchange of design information, and management of design change. No information is currently available, to our knowledge, on efforts to create active building components that can react to design changes.

## A MODEL FOR COORDINATION OF DESIGN INFORMATION

The detailed design stage of a building project is characterized by the processing of a large amount of information that change continuously. As a result, compatibility errors are commonly embedded in this information. Most of these errors are eventually discovered during construction and result then in change orders, contractual disputes, cost overruns, time delays, compromise to quality, frustration and client dissatisfaction. An information model has been proposed at the Centre for Building Studies to overcome this problem. The overall objective of the model, which is currently under implementation and testing, is to ensure adequate accommodation of design changes throughout building information at the detailed design stage, hence, reducing the occurrence of costly incompatibility errors.

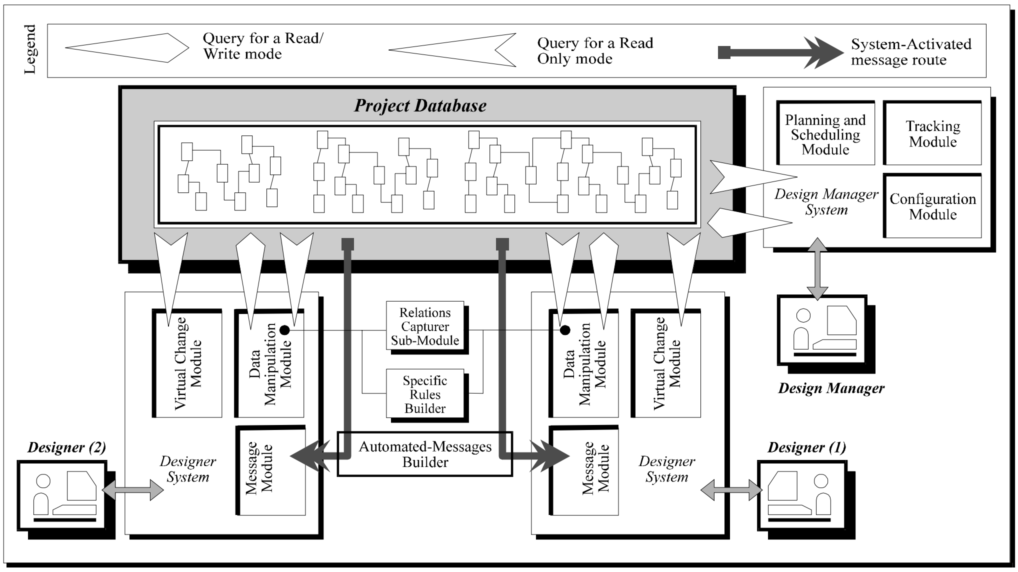


**Figure 1 Architecture of the Proposed Information Model**

The architecture of the main components of the proposed model is shown in Figure 1. The model integrates a central project database, designers from various disciplines and the design manager of the project. The central project database contains the project design information. A key feature of the project database is its ability to act as an active repository of building components. This feature allows each component in the building to recognize which design disciplines are affected by changing one of its specific attributes. For example, the window component "knows" that a certain change in its area, which is an attribute defined by the architect, may affect the HVAC system, which is defined by the HVAC designer. To achieve that, the building components are capable to check some rules and determine whether these are applicable for a certain design change or not. These rules state relationships that are necessary for design coordination. Two types of rules are used. The first type, the general rules, are applicable to the majority of buildings. An example for these rules is the fact that a change in the thermal resistance in the building envelope effects the HVAC system design. The second type of rules includes those that are specific to a certain project. For example, the height of the prefabricated elements in the building envelope equals double the floor-to-floor height. The general rules need to be pre-built with the building components while the specific rules need to be continuously captured from the designers and dynamically built with the building components. When any of these rules applies to a certain design change, the components automatically send messages to the concerned designers through the System-Activated Message Route. More details on the development of the information model are given elsewhere (Mokhtar et al. 1996).

## THE PROPOSED METHODOLOGY

In this section, we focus on the design of the central project database. That design needs to fulfill two main requirements. The first requirement is to make the database flexible enough to accept the design information of any building. That is an important requirement due to the fact that buildings are unique products. The second requirement is to have these data active in managing design changes.

## *Providing flexibility*

To fulfill the flexibility requirement, the relation database technology is utilized. This technology allows dynamic and transparent manipulation of data structure and data values by an interface program using the Structured Query Language (SQL). With such technology, each building component (e.g. window, beam, duct) is represented as a table. Each table has a different structure that represents the properties describing each component. The entities of a building component (e.g. window001, window002) are stored in their own specially designed table (e.g. window table). An interface program is developed to take advantage of the SQL language in adding, deleting and modifying tables that represent building components in the relational database. For example, if the architect decides to use circular windows in addition to the rectangular windows in a project, s/he will not be able to use the rectangular windows table to record the data of the circular window. That is because a circular window requires an attribute “Radius” instead of the attributes “Width” and “Height” of a rectangular window. Using the SQL, a new table for the circular windows is created with its own specific attributes in the relational database. The SQL statements for this operation include the following statements:

ìCREATE TABLE [Circular_Window] ([Circular_Window _id] TEXT(50));î

ìALTER TABLE [Circular_Window] add [Radius] NUMERIC (50);î

The interface program creates these statement automatically, the user needs only to provide the name of the building component and describe it through its properties. S/he does not need to know how this data is structured in the project database. The created table can now carry the data of all the circular windows in the project and in a similar fashion, any general rule can also be added to this new component.

Yet to enable the use of the SQL capabilities with an interface program, the concept of a management database is introduced. The management database keeps data that describe the structure of the tables that carry the building components data. With the introduction of the management database, the project database shown in Figure 1 is in fact composed of two parts. The first part is the *building components database* which is a group of tables each representing a building component (e.g. window) and carrying the values for the entities that belong to the component (e.g. window001, window002). The second part is the *management database* which is a group of tables that carry data about the tables structure in the first part (Figure 2).

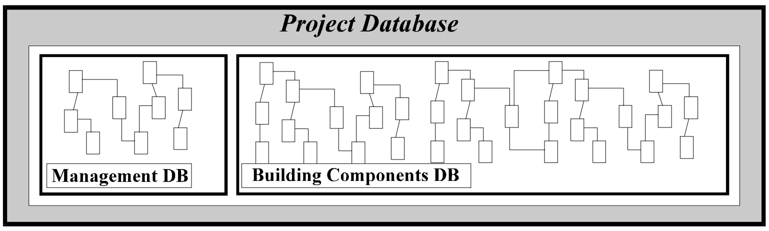


**Figure 2 The project database consisting of two components: the management database and the building component database**

## *Making the building components active*

The second requirement for the design of the central project database is to make the building components active in managing design changes. To fulfill that requirement, the Object-Oriented (OO) technology is adopted. In this technology, an object is a collection of related variables and procedures (Taylor 1990). Objects encapsulate methods, which are groups of procedures. When a building component (e.g. window) is designed, its properties (e.g. type, area, thermal resistance) are retrieved into an OO environment. Each property temporarily is represented as an object and consequently becomes encapsulated with a method. The method is triggered when a design change occurs to the property. The method retrieves some rules that are related to the property and checks their applicability to the design changes. If any of the rules is applicable, the method automatically sends messages to the affected designers informing them about the change and asking for an appropriate response (Figure 3). The rules are stored in a specially developed table that is part of the management database.

With such a combination of the relational database and OO technologies, only the relational database structure needs to be modified so as to accommodate the requirements of a project. Building components can be added and deleted. Their properties can be also changed. Rules can also be added or modified in the relational database.

### *Assembling the database structure of a new project*

Creating active building components in the project database and providing flexibility and manageability to the data structure of these components makes the proposed information model easier to implement in design offices. Yet, another difficulty still exists that is the large number of building components that need to be assembled in the database for a new project along with their rules. This appears to be a time consuming task that may discourage the use of the model. A general database therefore has been developed to facilitate this task (not shown in Figure 1). Three main observations influence the definition of the general database:

- The first observation is that building components are repeatedly used. As most of the building components are manufactured, they have to be widely used to reduce their cost. Nevertheless, only a limited number of the available building components are used in a certain project.
- The second observation is that building components can be grouped into systems. For example, a column and beam structural system will contain components such as concrete foundation, concrete column, concrete beam, and concrete slab.
- The third observation is that many buildings resemble other buildings in most of their building components . As example, most of the wood construction homes will be similar in their components .

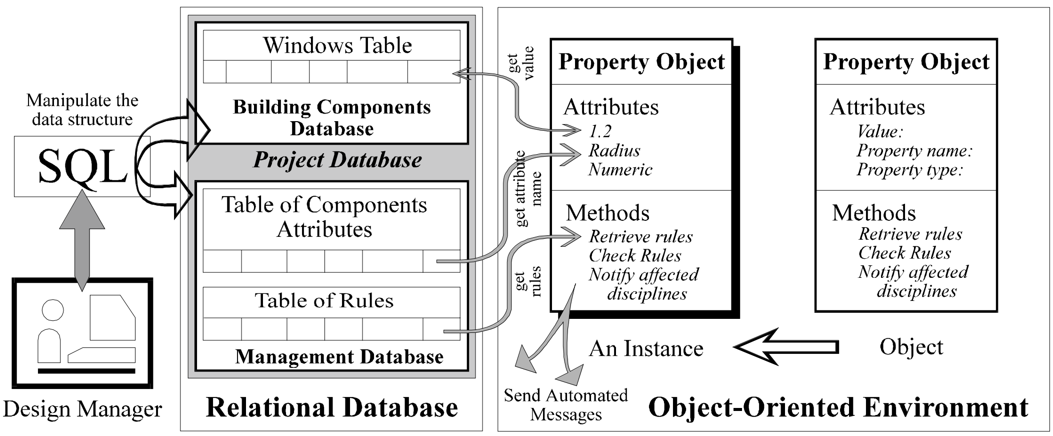


**Figure 3 Combined object-oriented and relational database technologies**

The general database design reflects these observations through assembling building elements into three levels of increasing complexity: Components, Systems, and Configurations. Members at the Components level are the primary ingredients in a building such as doors, beams, ducts, and luminaries. Each component is described through its properties. The next level is the Systems level. Examples for members of this level is cast-in-place flat slab concrete structural system, all-water HVAC system, and restroom system. Each system contains a group of related components that form the system. The restroom system, for example, includes lavatory, toilet, urinal, sink, plumbing, partition, mirror, etc. Each of these components is fully described in the components level. The highest level is the Configurations level. Members of that level are complete building configurations. Example are a concrete structure low-rise office building with all-air HVAC, or steel structure warehouse with infrared heating. Such configurations are required when a data structure for a similar project needs to be assembled.

As the General rules are usually valid for most buildings (for example, a change in the thermal resistance of an external wall affects the HVAC system design), they are considered reusable pieces of information. General rules are therefore reused, with some minor modifications, in every new project.

The general database is also implemented as a relational database with SQL capabilities to pick a component, a system, or a configuration from the general database and reconstruct it in the database of a new project. The structure of the general database illustrated in EXPRESS-G notation is shown in Figure 4.

## DISCUSSION ABOUT THE METHODOLOGY

Some researchers approached the problem of flexibility and extendibility of data models with the sole use of the OO technology. We see that this approach will ultimately face the shortcomings of this technology, mainly (Elmasri and Navathe 1994):

- No high level query language is available so as to allow the dynamic and transparent manipulation of data structure and data values by the end user through an interface software. The end user therefore needs to know how to use the development environment (usually an OO database management system). Such a requirement will severely limit the practicality of these models.
- Behavior rigidity, where a fixed set of methods has to be pre-determined and pre-specified during the model development, which makes it unfeasible to add and modify methods to building components during execution to suit specific project requirements.

We believe that our approach which combines both the relational database and the OO technologies provides a more practical solution, specially with the use of the general database to assemble the data structure of new projects.

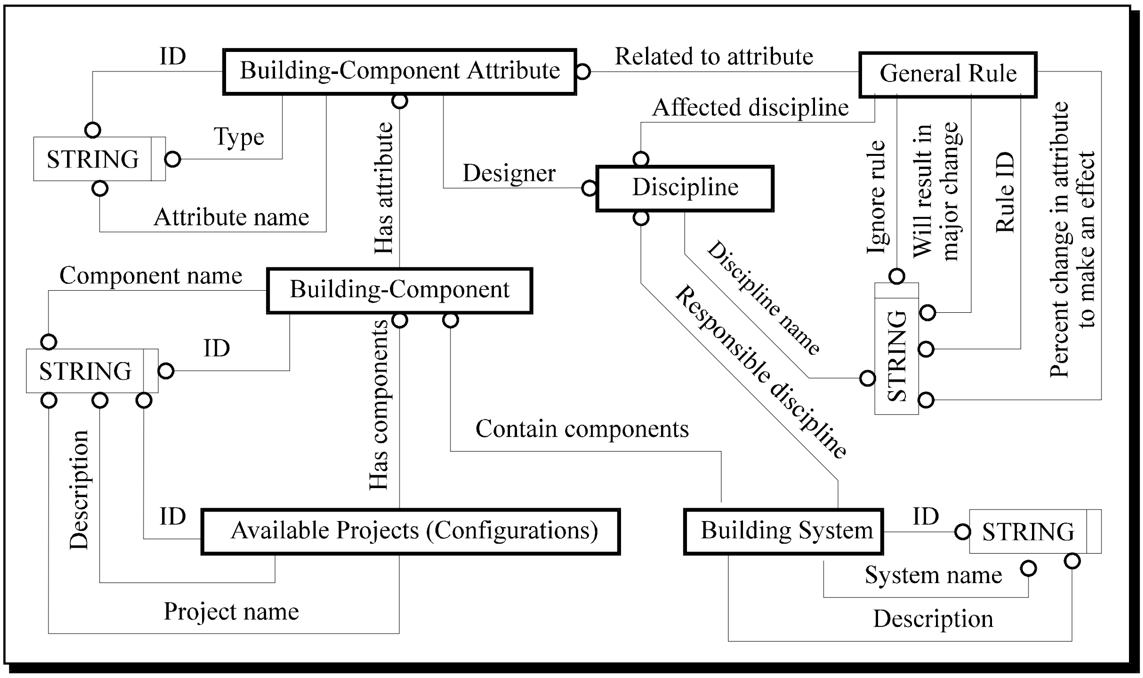


**Figure 4 The structure of the general database.**

## IMPLEMENTATION OF THE METHODOLOGY

To implement the developed methodology, the MS-WINDOWS (1994) platform is chosen because it is widely used in construction consulting offices. LEVEL5 OBJECT (1994) is used both as the interface software and the object-oriented environment that enhances the building components data with the object-oriented capabilities. The relational database is built on MS-ACCESS (1994) which is a relational database management systems. Both software have been linked using standard query language (SQL) through the ODBC (Object DataBase Connectivity) component of WINDOWS.

In a design office, the model should greatly help and raise the efficiency of the design manager who has the responsibility to ensure the compatibility of all design information before it is transferred to contractors in the form of technical documents. Yet, like most CAD systems, the model requires him/her to invest some time and effort at the beginning of using the model. Such effort is usually undertaken in cooperation with the disciplines involved in the project.

### *An implementation scenario.*

In this scenario, the design manager assembles the data structure for a warehouse project in order to start the detailed design stage of that project. All the main features of the warehouse have been previously defined through the former stages of the design process. By default, the model creates an empty new project database which contains only a structure for the management database part of the project database. As no data structure of a similar project has been assembled before, the design manager instructs the interface software that s/he will assemble the data structure using building systems (Figure 5). The design manager selects a design discipline (i.e. HVAC). A list of available building systems, that are the responsibility of that discipline, then appears on the screen as Figure 5 also shows. The design manager selects a system that is close in its building components to the preliminary design of the warehouse (i.e. all air system). Once s/he clicks on the 'add to the new project’ button, the structure of all the building components in the selected system is built into the database of the new warehouse project. New tables are therefore created in the building components part of the project database. The design manager continues to select other systems from the general database and builds them in the warehouse database. When all the needed systems are selected, then s/he clicks the finish button which triggers the system to select from the general database all the general rules that are related to the current components and add them to the warehouse database.

The system then provides the design manager with the option to further fine tuning both the data structure of the building components and the rules. As Figure 6 shows, the design manager can create new components, modify the structure of current components, or delete component that are not required. The figure shows that the new component “Circular_Window” is being created along with its properties. General rules may also be added and current rules can be modified or ignored as shown in Figure 7 where a general rule is created to let the system react when a change of 30% in the radius of the circular window occurs and a message should be sent to the HVAC designer.

The design manager is able to transparently do all these actions without having any knowledge of the development software (MS ACCESS) because of the existence of the management database and the utilization of the SQL language in the interface software.

Once the data structure of the warehouse is created, designers from the various disciplines can now populate the database with design information. For example, the architect creates a circular window “c_window001” and fills its properties with design information (Figure 8). In a later stage of the development of the detailed drawings, the architect changes the “radius” of that circular window from 1.2m to 1.8m (Figure 9). The method attached to the attribute “radius” is automatically triggered and it retrieves from the relational database all the relevant rules to the “radius” of the circular window. One of these rules indicates that the HVAC designer need to be notified when a change of 30% occurs in that attribute. As such, a message is sent automatically by the “circular window” itself to notify the HVAC designer (Figure 10). So even if the architect is not aware that any other discipline might be affected by his/her design change, the changed building component itself takes care of notifying other disciplines. The change therefore has much better chance now to be accommodated and the design information is less likely to contain incompatibility errors.

## CONCLUSION

A central database that carries the design information of building components has to be flexible in response to the fact that buildings are unique. The task becomes more complicated when such building components need to be active in notifying about design changes. A combination of object-oriented and relational database technologies has been utilized to achieve the required flexibility. The central database has been also divided into a management data part and building components part. The management data part carries data about the structure of the tables that reside in the building components part. It therefore allows the use of the SQL capabilities to dynamically and transparently modify the database through an interface program. In addition, a general database has been designed to carry the reusable elements in building projects. From these elements, the database of a new project can be assembled more easily and rapidly. The methodology has been successfully implemented and tested through an illustrative scenario.

## REFERENCES


Abou-zeid, A., Russell, J., Hanna, A., Park, S., 1995, "Data Flow Model for Communications between Project Participants in a Highway Bridge Project," Canadian Journal of Civil Engineering, Vol.22, pp.1224-1234.

Augenbroe, G., 1992, "Integrated Building Performance Evaluation in Early Design Stages," Building and Environment, Vol.27, No.2, pp.149-161.

Bédard, C., Rivard, H., 1995, "Integration in Building Engineering Design with Database," Computing in Civil Engineering, ASCE, pp.502-508.

Björk, B-C., 1989, "Basic Structure of a Proposed Building Product Model," Computer Aided Design, Vol.21, No.2, March.

De Waard, M., 1992, "Computer Aided Conformance Checking: checking residential building designs against building regulations with the aid of computers," Doctoral Dissertation, Delft University of Technology, Delft, The Netherlands.

Dubois, A.M., Parand, F., 1993, "COMBINE Integrated Data Model," CIBSE National Conference, pp.96-108.

Eastman, C.M., Cho, M., Jeng, T., Assal, H., 1995, "A Data Model and Database Supporting Integrity Management," Computing in Civil Engineering, ASCE, pp.517-524.

Elmasri, R., Navathe, S., 1994, "Fundamentals of Database Systems," Benjamin Cummings, Don Mills, Ontario.

Herold, K., 1997, "Editorial-Universal Building Language," Journal of Computing in Civil Engineering, January, pp.1-3.

IAI, 1996, Industrial Alliance for Interoperability, http://www.interoperability.com

Karhn, V., 1997, "Product Model Based Design of Precast Facades," Electronic Journal of Information Technology in Construction - ITCon, Vol.2, http://itcon.org

Krishnamurthy, K., Law, K., 1995, "A Data Management Model for Design Change Control," Concurrent Engineering: Research and Applications, Vol.3, No.4, December, pp.329-343.

LEVEL5 OBJECT, 1994, Level5 Object for Microsoft Windows - Reference Guide, Information Builders, NewYork.

Mokhtar, A., Bédard, C., Fazio, P., 1996, "An Active Database for Inter-Disciplinary Coordination of Building Design Information," Proceedings of The Third Canadian Conference for Computing in Civil and Building Engineering, Montreal, Canada, August 26-28, pp 748-759.

MS-WINDOWS, 1994, Microsoft Windows - User's Guide, Microsoft Corporation.

MS-ACCESS, 1994, Microsoft Access - User's Guide, Microsoft Corporation.

Platt, D., 1996, "Building Process Models for Design Management," Journal of Computing in Civil Engineering, Vol.10, No.3, pp.194-203.

Rezgui, Y., Debras, P., 1995, "An Integrated Approach for a Model Based Document Production and Management," Electronic Journal of Information Technology in Construction - ITCon, Vol.1, http://itcon.org

Sadri, S.L., 1993, "An Integrated Information Model for Managing the Design and Construction of Building Construction Projects," Proceedings of the First International Conference on Management of Information Technology for Construction, World Scientific and Global Publications Services, Singapore.

Svensson, K.E., 1993, "Maintenance Information Management, Based on a Step-Related Building Product Model," Management of Information Technology for Construction, edited by Mathur, K., Betts, M., Tham, K.W., World Scientific and Global Publications Services, Singapore, pp.509-516.

Taylor, D., 1990, "Object-Oriented Technology: A Managers' Guide," Addison Wesley Publishing Company.

Tsou, J-Y., 1992, "Using Conceptual Modelling and Object-Oriented Environment to Support Building Cost Control during Early Design," Doctoral Dissertation, University of Michigan, Ann Arbor.

Turner, J.A., 1990, "AEC Building Systems Model," ISO TC184/SC4/WG1 (STEP) working paper.

Turner, J.A., 1992, “Conceptual Modelling Applied to Computer-aided Architectural Design,” Building and Environment, Vol.27, No.2, pp.125-133.
Vries, B., Somers, L., 1995, “Message Exchange in the Building Industry,” Automation in Construction, Vol.4, pp.91-100

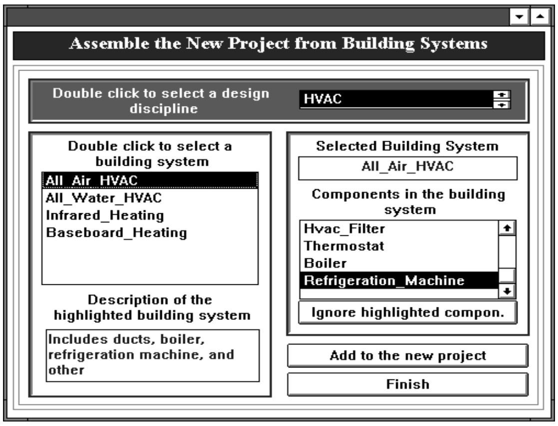


**Figure 5 Interface to assemble the data structure from the building systems available in the general database.**

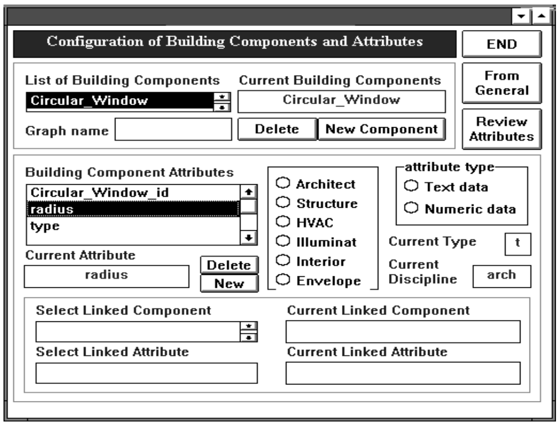


**Figure 6 Interface to fine tune the data structure of the building components of a project.**

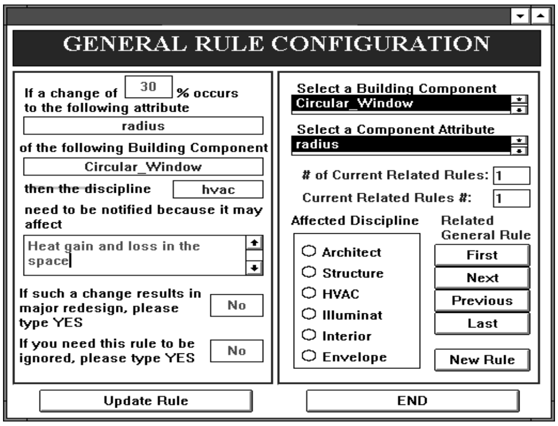


**Figure 7 Interface for manipulation of general rules**

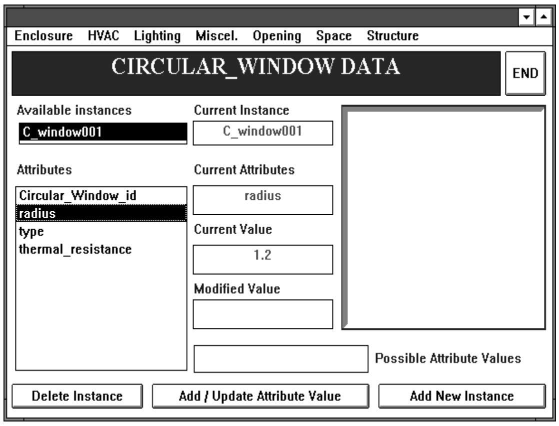


**Figure 8 Add design information of a building component by architect**

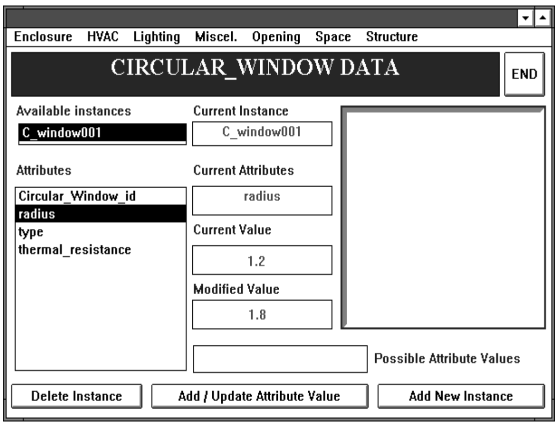


**Figure 9 Change design information of a building component property by architect**

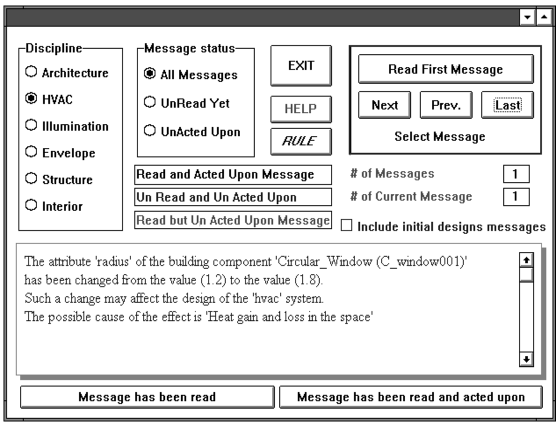


**Figure 10 A message that is automatically sent by the changed building component to HVAC designer**