Integration of Knowledge and Data: Results of the BECOC Prototype

I. Bellot, C. Barrobés, A. Isamat, M. P. Belinchón and A. Martinez[1]

Abstract

The BECOC prototype - "Banc Estructurat de Coneixements sobre Elements Constructius" (Structured Knowledge Bank for Construction Elements) - is the first practical step to have been taken in the research project of the SITEC model. It is an example of a system designed to provide assistance in the control of standards in the project process of building design. More specifically, it controls the acoustic, thermal and gravitational standards, and is centered on the design of a building's shell. This prototype has represented a considerable advancement in the research on integrated modular systems and on knowledge representation systems in applying Artificial Intelligence techniques.

Introduction

The BECOC prototype is an assistance system in controlling due compliance with acoustic, thermal and gravitational standards during the process of assigning constructive solutions to the shell of a building. It has validated the SITEC model (Bellot, 91; Costa & Valencia, 87; Costa, 91; Guilera et al, 89) concerning information treatment on alphanumeric and graphic data and knowledge that has been treated in a decisive and at the same time modular way, integrating different subsystems within a single common structure.

It has been the first practical step taken within the research project of the SITEC model. One of the objects of the project was that of consolidating a model of abstract and yet at the same time well defined representation, capable of covering the wide range of knowledge involved in any design project in the construction world and on any level of complexity.

The BECOC prototype has integrated a traditional Relational Data Base (for the massive treatment of the alphanumeric information), an Expert System and a graphic module with parametric treatment and object oriented graphic information. The structure of the information has been unique and common for all the modules or subsystems (Fig.1).

It has been proved [6-7] that the means of technological solution for solving the problems or the basic necessities presented nowadays in any construction project (during its life cycle) is moving in a direction where the treatment of information is object oriented. Nevertheless, it is unavoidable to add a treatment of dynamic, parametric and deductive knowledge which can no longer be resolved merely with traditional systems based on relational frameworks, but integrated within systems based on reasoning and generic representation and standards of knowledge.

In the BECOC prototype, knowledge has been represented by means of an Object Oriented Data Base and a Rule System. The O.O.D.B. comprises static elements (classes, objects, properties and metaslots) that represent the initial information within the Data Base, and dynamic elements (objects with inheritance of properties and metaslots) that are generated in

[1]Institut de Tecnologia de la Construcció de Catalunya - ITEC Barcelona, Spain

Construction Informatics Digital Library http://itc.scix.net/
paper w78-1992-46.content

real time, treated as persistent objects and which represent the information on the problem in hand.

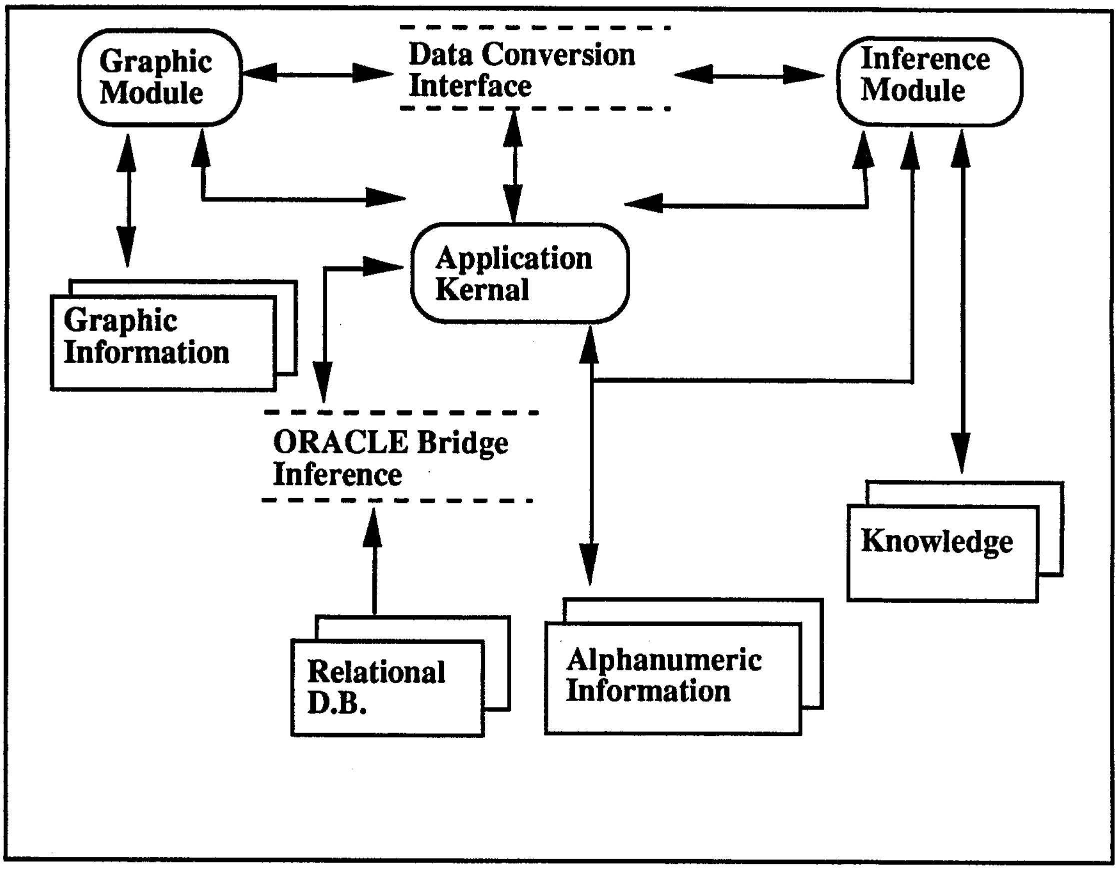

Figure 1 The BECOT Information System

The Rule System includes the inference engine that triggers off the rules and acts deductively on the objects, thus providing dynamic information.

Functional Division

Given the complexity of the project, its study and implementation has been divided into specific functions (Fig. 2), which are:

1. The acquisition of the initial information required for the due consideration of the problem (initial conditions).
2. Formal definition of the project (initial geometric scheme).
3. A decision taking system based on data and knowledge.
4. The design of a generic model for representing Data and Knowledge, using Expert Systems technology and Object Oriented Data Bases.
5. Design of Front-End capable of integrating different modules, of acquiring user input information, and presenting the context (state of the information) resulting from the continuous reasoning process, at each moment.

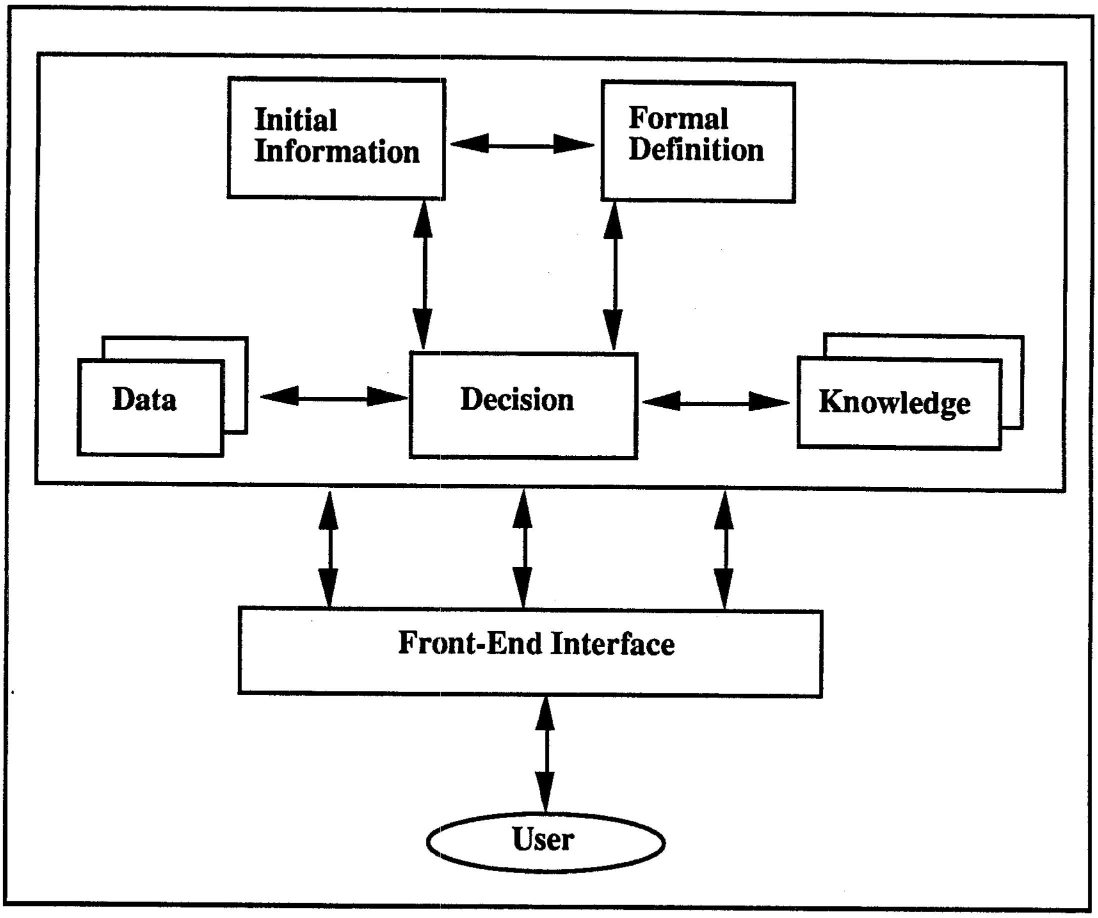

Figure 2 Specific Functions in the BECOT prototype

The functionalities as mentioned above have been grouped into:

Reasoning module: This includes points 3 and 4. Decision taking system and the generic module of representation of data and knowledge developed with NEXPERT/OBJECT.

Front-End Interface: This includes points 1 and 5. User-oriented Front-End and initial information acquisition by the system, developed within the X-WINDOWS framework.

Graphic module: This includes point 2. A system prepared for drawing parametric objects in hierarchy, developed with the STARBASE library and "C" language.

All the modules are integrated using "C" language, in that moment considered to be the most suitable one.

The advantages this sub-division represents are several:

- It makes expansion of the SITEC model easier, both in structure and applications: the integration of a new structure module or new application (e.g. a module for optimization of the result obtained, a module for graphic output, amongst others) will not be difficult.

- It makes it easier to maintain parts of the model: it will be easy to modify a module without affecting any other. E.g. it will be easy to change the presentation of the Front-End without affecting the system's reasoning module.

- Transportability of information between different applications: by a common structure and with a standard format, it is possible to transfer information from graphic design applications to others, such as applications on quality control, and generation of price tables, amongst others.

Reasoning Module

In the BECOC prototype, two types of information are available: data (concrete information) and knowledge (abstract and conditioned information). Data and Knowledge are integrated by means of the Expert System provided by an Object-Oriented Data Base and a Rules system. The inference engine that forms a part of the reasoning system triggers off the rules on the objects.

The prototype demonstrates the competence of the expert system in representing knowledge of thermal, acoustic and gravitational standards which, when applied to the project being carried out, is capable of controlling and regulating the factors involved in the building.

The Object-Oriented Data Base is formed by static information: it contains generic and parametric information on the construction elements comprising a building's shell (the walls, ceilings, roofs, hearths) and information on the simple elements and work units modeled in the prototype. This information is represented by means of classes with properties and specific defined actions (Fig. 3).

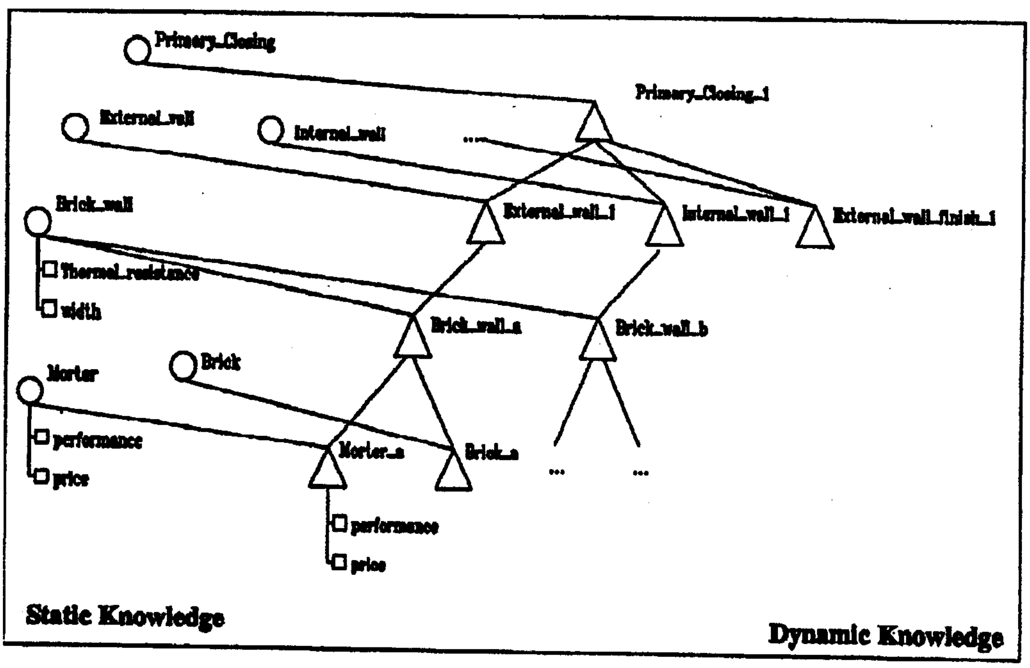

Figure 3 General Knowledge Bank Design

Dynamic information is generated during the design of a specific construction project: the objects representing graphic elements defined in the formal definition will be dynamic objects. These objects are instanced into classes (static generic information) to inherit their characteristics and their environment. An object is therefore an instance within a class. Data deduced during the reasoning process, and those brought forth by the user, are also added as dynamic information on the structure.

Due to the volatile nature of dynamic information, the dynamic objects are treated as persistent objects. At the end of the session, all the information forming a part of a particular project carried out is stored.

During the design process of a construction project, the number of elements that can be used is very high. This means counting on a data base of very large dimensions in order to treat and store the information, or else sub-dividing the information into different data bases depending on their type. Since the expert system used (NEXPERT/OBJECT) does not have a data base for dealing with large volumes of information, it will be necessary in future applications to be availed of a data base external to the expert system yet integrated with it (Bellot & Dedes, 92).

However, the volume of data involved in the BECOC prototype (and that will intervene in any application of the SITEC model) has been considerably reduced, due to the use of parametric information: the information is represented by using basic data and knowledge relating such data (and their associated information) in order to obtain new data by means of the reasoning process being produced in real time (Bellot & Dedes, 92).

Direct connection has been made between the Rule Conditions System and the user Front-End in order to establish dialogue between the user and the expert system itself. This connection is made by means of request calls from the expert system rules to external auxiliary procedures developed in "C" language.

The prototype was developed in "C". The reason for this was the need of an open language that would allow integrating different environments or software. An object-oriented language of a "C++" type was not considered necessary since all the management of the objects and inheritance was carried out within the expert system in an adroit manner.

Graphic Module

The graphic system developed is in charge of collecting up the geometric information on the shell of a building in the best user-oriented manner possible.

This information is to be provided by an expert user, architect or a designer, for whom a suitable graphic framework has been designed. It is for this specific reason that the graphic system is based on a set of menus structured on different levels (tree structured) and on a series of windows for presenting relevant data and for visualizing at all times the graphic state of the project.

The CAD system is thus divided into the following sub- systems:

Menu sub-system: This facilitates dialogue or interaction between the user and the CAD process. The menu structure is based on a descendant hierarchy of menus formed of N levels in depth. Branches within the tree structure have graphic actions associated to them, whereas the intermediate nodes are sub-menus. The user is availed of a series of icons on the screen that are representative of the level of design in which he is situated.

Design sub-system: This consists in a series of operations that are controlled by the menu sub-system. It contains the management process of the structure of the graphic system's internal data, the control processes of the pre-established restrictions such as spatial orthogonal features, generation of a base floor and N typical floors, and the necessary routines for calculating - amongst others - the volumes and areas.

Visualization sub-system: This is comprised by graphic windows visualizing the information output. Each window represents a view, a floor, an elevation or a perspective. Furthermore, there is a window for drawing.

Interface with the Expert System: This is a set of operations which, as from graphic structures, provide essential graphic information for the reasoning of the expert system (areas of the elements, elevations, widths, and the types of separation between premises). The menu sub-system provides the facility of modifying, deleting or adding options associated to new icons. Furthermore, it allows the user to work in a comfortable and simple manner.

As there is graphic information separate from the Expert System, changes in this information do not affect the rest of the system whereas any change in the shared graphic information represents an up-dating of the internal structure of the expert system.

Layout of the windows on the screen is stable. Although this may appear to diminish the flexibility of the user's work, this is not so since the views contribute in an associate manner to direct actions of zoom, translation and rotation of the drawing, and one may also move over into the drawing window.

Communication Font-End

The user interface developed in the X-WINDOWS framework allows integration of the remainder of the modules.

The X-WINDOWS system is object-oriented where the treatment and structurization of the windows are concerned. Concepts of super classes, classes, sub-classes and inheritance appear within the behaviour and aspect of the windows, and in the events being produced (Winblad et al, 90).

A hierarchy has been constructed (a window-tree) that represents the user's interface of the application. Different types of windows appear according to the requirements: a basic window (layout, bulletin board widget), menus (Menupane widget), dialogue (Textedit), and data presentation (Forms, Row-columns, Scrolls).

It is very important for the user to be availed interactively of the maximum accessible information at the moment of designing the construction project; this availability provides the user with a fast and comfortable means of working.

The external application is based on a series of navigation menus and final nodes for gaining access to the screens that correspond to the different phases of design of a project.

In another sense, the programming mode in X-Windows is event driven, meaning that the application is more user-oriented: it is always the user himself who decides what he wants to do.

Any alteration on behalf of the user of the data concerning the project brings about an up-dating of the dependent information. In this manner, in real time, changes that occur in the expert system reasoning discourse are visualized.

In summary, the main functions specified in the BECOC prototype under the X-WINDOWS framework have been:

- Definition of the window hierarchy structure (front-end aspect).
  Callbacks: Functions that trigger off as a result of an event produced on a window.
- Communication with other modules. Request calls to the functions of the graphic module and the reasoning module.
- Interaction functions with the user: management of dialogue windows (data input).
- Information presentation functions: management of up-dating and data output.

Use of windows gives the application an ergonomic aspect: data input is agile; it is based on the choice of values amongst a series of options; information is viewed in a structured and coherent manner, showing the most relevant pieces of knowledge on the screen.

Integration

Upon having made the graphic and reasoning modules as separate as possible, the need arises for each module to contain all the structures with all the information relevant to it, so the information shared by the two modules is found "duplicated". Each one of these modules shapes a different aspect of this information: in the graphic system it will be most important to know not only the area, elevation or width, exact situation and coordinates of a window, whereas for the expert system what will be essential are the parameters such as area, height, width and type of separation, amongst others.

Structure of the Information

Part of the initial information required in the BECOC prototype is the geometry entered into the system via the graphic module. Only part of this geometry is used later by the expert system.

The expert system needs relatively little information, and this can be defined basically in a list of all the objects of graphic elements created (taking as objects, on this level, elements to which a construction solution will later be assigned: the roofs, walls, windows, ceilings, hearths, doors) plus some dimensional data (volumes and areas) and positional information (upon which wall a window is to be found, the type of separation of an element, whether or not there is heating).

Thus, output of the graphic process is concreted in a list of simple instructions to be interpreted by the expert system. These instructions are:

- Create a new object with the associated graphic information, together with its identifying elements when a graphic element is created.

- Delete/modify an object already created. This means to say deleting or modifying a graphic element or any information associated to it. The BECOC prototype makes it possible to exit the reasoning process to return to the graphic system to modify/delete objects, and then to later continue with the reasoning process.

When the drawing session comes to an end, a file is generated containing the list of instructions that will later be read and interpreted by the expert system. Once the information has been gathered up the expert system takes charge of removing the file.

The advantages of this implementation are:

- It facilitates separate development of the geometric and reasoning modules, also facilitating the independence or possibility of substitution of the latter (e.g. by any graphic system available on the market).

- Unnecessary data transfers are avoided. Only the geometric information validated after a work session with the graphic module is transferred. (Information on objects created and deleted during a single session is not transferred).

Evidently this means that upon finishing a graphic session there is a high flow of information to be transferred to the expert system. The reasoning system will have to create many objects at the same time and this slows down execution at a given moment (it is considered that the average number of elements a building can have is of around one thousand).

Also studied was the way to avoid duplicity of the information by centralizing it in the expert system, so that each creation and modification of graphic objects would be translated directly upon a request call to the expert system. This alternative was rejected since it involved certain serious inconveniences:

a) Overload in the flow of information between both modules.

b) Continuous interaction between the modules, which made the separate development of the two modules by different teams of people difficult, and which meant that the system was too intimately bound, making it difficult to substitute any of the parts without affecting the other;

c) The mixing of information of different semantic content: the expert system would hold geometric information that was not useful mixed with the relevant information.

Modularity

Modularity and independence of each part has allowed taking maximum advantage of the software used:

- The X-Windows framework allows working in a single session in parallel with different windows, with the modules of the same application, or with different applications that interchange data.

- Full advantage has been taken of the expert system's capacity to design O.O.D.B. and to represent knowledge on a shell in a particular but simple language Turner, 92).

- A graphic module has been developed which, at any moment, may be substituted by any other on the market.

As a result, integration has been based on passing messages from the two modules to the front-end and vice versa; in the specific case of the expert system, the messages are triggered off from the rules themselves.

Conclusions

In view of the results obtained with the BECOC prototype, the study and continuity of the SITEC model is addressed towards:

- The consolidation of the SITEC structure, based on the treatment of parametric data and knowledge, on varying levels of complexity.

- Continuity in research of systems representing knowledge, in accordance with the new trends or techniques of Artificial Intelligence.

- Research of a tool for developing Front-Ends of standard applications, transportable between different shells.

- Research on CAD systems with parametric treatment, object- oriented, on a 2D and 3D level, that can be integrated with other sub-systems (both graphic and alphanumeric).

In summary, implementation of the BECOC prototype has been of assistance in going further in depth in the consolidation of the SITEC model, and has opened new ways in the search for CAD systems that, in future, will make it possible for computers to provide "intelligent" support to architects and designers within the different fields in construction.

References

[Bellot 91] Bellot Isabel; "BECOC: "A Knowledge Bank and its use in Construction and CAD Systems", CIB W78 Seminar, September 1991 Eindhoven - The Netherlands

[Bellot 92] Bellot Isabel and Christos Dedes; "The SITEC model as a Knowledge Based System : Initial phase implementation.", First International Symposium of CIB Working Commission W82: Future studies in Construction, ESPOO, June 1992, to be published

[Costa 87] Costa Felip and Valencia Eladi; "Una Base de Dades amb Estructura Generadora d'Elements Constructius", ARECDAO'87 Proceedings, 1987 Barcelona, pp.112-119

[Costa 91] Costa Felip; "El Tractament dels Coneixements i el seu ús en les aplicacions per a la construcció. Estat de l'art i resulats d'una experiència", ARECDAO'91 Proceedings, 1991 Barcelona, pp.59-67

[Guilera 89] Guilera Llorenç, Mejino Llorenç and Costa Felip; "Sistema Integral de Tractament d'Elements Constructius", ARECDAO'89 Proceedings, 1989 Barcelona, pp.87-100

[CIB W78+W74] Proceedings of the CIB Seminar "Computer Integrated Construction", CIB W78+W74, Tokyo, September 1990

[CIB W78] Proceedings of the CIB Seminar "The Computer Integrated Future". CIB W78, Eindhoven, September 1991

[Turner 92] Turner Scott R.; "NEXPERT Object, IEEE Expert, March 1992, pp.72-77

[Winblad 90] Winblad Ann L., Edwards Samuel D. and King David R.; "Object Oriented Software", Addison Wesley, September 1990.