CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Linking Knowledge-Based Systems to CAD Design Data with an Object-Oriented Building Product Model

by Kenji Ito, Yasumasa Ueno, MASCE,
Raymond E. Levitt, MASCE, and
Adnan Darwiche

TECHNICAL REPORT
Number 17

August, 1989

Stanford University

Linking Knowledge-Based Systems to CAD Design Data with an Object-Oriented Building Product Model[1]

Kenji Ito[2], Yasumasa Ueno, MASCE[3], Raymond E. Levitt, MASCE[4], and Adnan Darwiche[5]

Abstract

Sharing of product data among discipline specialists and across stages in the product life cycle is difficult in many industries, but particularly so in the highly fragmented architecture-engineering-construction (AEC) industry. The industry currently lacks accepted standards for product data interchange which might allow applications to share the many kinds of graphical and non-graphical data produced and needed in the various stages of the AEC process. As a result, existing practice is to hand-craft interfaces between any given pair of applications at great expense and with chance for re-use. Object-oriented programming in which objects inherit attributes – describing both their state and their behavior – from more general objects in a "frame" or abstraction hierarchy, offers a powerful new approach to sharing information and knowledge in the AEC process.

This paper covers the development, present status and future directions for PMAP, an object-oriented building project model developed to facilitate the sharing of information between participants in the design, construction and operation of buildings. We describe a high level user interface developed in AutoCad to capture building component data; we show how this data is then transformed in two steps into an object-oriented building product model; and we demonstrate the feasibility and practicability of using this type of object-oriented product model for information exchange by linking it to two separate knowledge-based systems for generating construction project plans.

Keywords:

CAD; object-oriented programming; building models; integration; decision support; database management systems; artificial intelligence; expert systems;

---

1 This research was funded by seed research grants from the *Center for Integrated Facility Engineering* , Stanford University, the National Science Foundation Grant # MSM-8716608, and the US Army Construction Engineering Research Laboratory. The authors are indebted to these organizations for their support of the research.

2 Senior Research Engineer, Shimizu Corporation and Visiting Fellow *Center for Integrated Facility Engineering* , Stanford University

3 Senior Research Engineer, Kumagai Gumi Co.,Ltd. and Visiting Fellow *Center for Integrated Facility Engineering* , Stanford University

4 Professor of Civil Engineering and Associate Director, *Center for Integrated Facility Engineering* , Stanford University

5 Research Assistant, Departments of Civil Engineering and Computer Science, Stanford University

1. Introduction

The construction industry is highly fragmented both across disciplines and across stages in the product life cycle. Specialist individuals and firms have successfully used computers to automate and aid much of the work within their own narrow areas of specialization, but have thus far found it very difficult to share data in machine-readable and machine-usable form across the industries myriad boundaries.

Many of the reasons for this "failure to communicate" are institutional rather than technical. [Howard 1987] discusses some of these reasons, which include concerns about liability for errors or omissions in machine-readable files issued by their organizations, as well as concerns related to the ownership of intellectual property in the amalgamated data and knowledge base of a completed facility. These are clearly important barriers to further integration of data and knowledge in the industry, but are not the subject of the research described here. The authors are part of a team of researchers in Stanford's *Center for Integrated Facilities Engineering* (CIFE) concerned with developing advances in information technology to facilitate integration and automation in the industry. Other researchers in the Center are conducting research on organizational and institutional means to the same end.

This paper describes the development and use of an object-oriented building project model as a vehicle to share product information across a range of computer applications in the AEC product life cycle. Specifically, we describe a high level user interface that we developed in AutoCad using AutoLisp, which facilitates the capture of building component data during the design process. We show how this data is transformed in two steps into an object-oriented building product model which can be used to drive a range of knowledge-based system applications in the AEC process. We demonstrate the feasibility and practicability of this object-oriented product model by linking it to two separate knowledge-based systems for generating construction project plans. We describe problems encountered in the development of the PMAP building project modeling system, our strategies for addressing these problems, the current status of the system, and our goals for extending it.

2. Related Work

2.1 Sharing data between applications

In many industries, companies have selected their own CAD systems and applications and have developed specific ad-hoc interfaces between them. Many of these systems have been utilized effectively within narrow application areas to form local "islands of automation." However, in the Architecture/Engineering/Construction (AEC) process, it is usually more productive to integrate information along the paths through which work flows between project stages or specialists – i.e., from the planning stage to the design, construction and management stages of a facility, or from architectural to structural and mechanical design.

To support data-exchange between different computer applications, several standards have been developed in an attempt to facilitate data interchange. Examples of such attempts at committee-defined or *de facto* industry standards include IGES, and DXF, respectively, in the case of graphical data. However, both of these standards impose restriction on the data types and formats which they can exchange. Hence, direct exchange of graphical data via customized one-to-one interfaces is still frequently employed by firms in the fragmented AEC industry in an effort to avoid tedious manual reentry of data across diverse applications, a process which has been referred to as "UN-CADD" [Howard 87].

In order to solve these problems more effectively, several national and international attempts are ongoing to develop richer product data exchange standards which can be used to communicate non-graphical as well as graphical data across applications, particularly to aid computer-integrated manufacturing (CIM). ISO(International Standards Organization)/STEP(Standard for the Exchange of Product model data) is the first advanced effort of this type by European countries. The PDES(Product Data Exchange Specifications) is a US effort initiated because of widespread recognition that IGES is an inadequate solution for defining and exchanging product design data. PDES is positioned as a successor to IGES, and as a standard that can generate inputs to ISO/STEP [Warthen 88].

AEC participants have been involved in these committees and are providing input about the needs of the industry for computer integrated engineering and CIM [Geilingh 88]. Yet progress in such committees is invariably fraught with political and economic overtones that delay the evolution of standards. The tendency has been for the data storage formats of the most widely used products (such as Lotus 1-2-3 or AutoCad) to become *de facto* data interchange standards to and from which other applications can convert their data. The AutoCad DXF format is currently close to becoming this type of standard for graphical and limited attribute data in the US AEC industry.

2.2 Product modeling in the AEC process

Several researchers are attempting to develop object-oriented product models which could be used to carry richer kinds of data and knowledge about a product, including knowledge about its design, manufacturing and operational parameters. Prof. Kincho Law of Stanford University is carrying out research on structural design using an Object-Oriented-Product-Model(PENGUIN) [Law 1987], [Wiederhold 1986], [Law 1989] The Technical Research Centre of Finland has been making progress in multiple studies about a building product-model (RATAS) using relational databases and hypermedia[Pentilla 1989].

Prof. Robert Logcher of MIT is working on an object-oriented CAD system (BUILDER) which generates its symbolic model as the graphics is created on a LISP machine [Cherneff 1988] [Logcher 1989]. In this prototype CAD system, IntelliCorp's KEE system with its KEE-Pictures™ utility is utilized as a CAD-like user-interface to develop an interior layout. The data generated by KEE-Pictures becomes an Object-Oriented-Model within the KEE environment. The object model serves as input to a prototype system of knowledge-based planning and scheduling implemented in KEE.

BUILDER is closest in its objectives to the work described in this paper, but its implementation strategy is diametrically opposed to ours. We do not attempt to recreate a CAD system within an AI environment. Rather, we aim to use knowledge-based systems technology to integrate data and knowledge across industry standard CAD and Project Scheduling software packages – AutoCad and MicroPlanner in our case.

3. Our Approach

3.1 PMAP (Object-Oriented-Project-Model for AEC Process)

BUILDER is interesting in terms of its development of an Object-Oriented Product Model in an AI programming environment and it achieves a smooth integration between the CAD and planning functions using this product model. However, the software and hardware environment of BUILDER – KEE on a Lisp Workstation – is still uncommon and expensive, and its CAD user interface is restricted to a few simple kinds of data. Furthermore, the BUILDER model does not include a general description of buildings, or even a specific description of the overall building for each project. This contextual information and knowledge is very important and useful in many other tasks associated with the AEC process (e.g., cost estimating, fire safety evaluation) that might be integrated with the planning that BUILDER does.

In our product modeling strategy, we have tried not to restrict the model's applicability to a specific aspect of the AEC process. Rather, we have defined a PMAP (Object-Oriented-Project-Model for the AEC Process) (Fig-3.1.1) which can be applied widely. Our model has initially been used as a vehicle to share product information for a mid-rise building project between a CAD system and two expert planning systems. The initial applications have involved information about the building's foundation, structural and architectural systems. We are in the process of extending the model to include mechanical system components and finishes.

3.2 CIFECAD (Customized AutoCAD for the AEC process)

We decided to use a CAD system which is in widespread use among many AEC companies as a user interface to our Project-Model. Our user interface was designed to enter both graphical and non-graphical information about building elements at the same time. Using this user interface language, we developed a prototype of a workable PMAP modeling system which can be linked to other applications. We selected AutoCAD as the CAD system interface to PMAP because of its widespread use in the AEC sector, and because of its ease of customization via the embedded AutoLisp programming language.

3.3 Other Applications Linked to PMAP

In order to demonstrate some generality for our product model, we chose to use it to supply product information to two separate knowledge-based construction planning systems – OARPLAN [Darwiche 1988], and SIPEC [Kartam 1989] which are currently under development at Stanford University.

4. Implementation of the Prototype System

We describe the steps in implementing the prototype PMAP model and a pair of interfaces from it to two knowledge-based planning systems in this section. In doing so, we point out some problems that were encountered and describe the strategies that we selected for addressing these problems.

4.1 The Environment of the PMAP System

In developing the prototype, we considered the overall information requirements of the AEC process so that we would be able to use our model as an interface to many different AEC applications. We wanted our model to be a Project-Model in the AEC process, not a Product-Model in the manufacturing sense. Our model should be usable at any stage from planning to facility management.

We selected the Macintosh II as the hardware platform for our prototype because of its excellent support for AutoCad, Common lisp and graphically oriented project scheduling tools.. We selected Allegro Common Lisp and Frame Kit for programming our project model. Allegro Common Lisp is the most popular AI-language on the Macintosh. Frame Kit, a frame-based knowledge representation shell developed at Carnegie Mellon University, is written to run under any Common Lisp and provides higher level functionality for manipulating classes of objects, including inheritance of attributes.

Our integrated system thus consists of the customized AutoCAD system (which we call CIFECAD) implemented in C, and AutoLisp and PMAP running under Allegro Common Lisp on a Macintosh II.

4.2 Problems with the Implementation

We encountered a number of problems during system implementation. They are described in this section along with the strategies that we developed to overcome them.

The user-interface and the database-interface for our CAD system posed two problems for us. First, AutoCAD has been developed up through Version 10 as a drafting system rather than as a design system. There is no high-level way to input building elements even in graphical form via the original AutoCAD interface, although several third party vendors have developed such intefaces to facilitate drafting in specific domains. Our planning application requires much more than a high-level graphic interface; we need a convenient means to enter and communicate several kinds of non-graphic or "attribute" information (including relationships to other elements) for each building element, in order to support our planning applications. Second , AutoCAD data are stored in a CAD database termed a *.DWG* file in binary form. Thus we could not access the AutoCAD data from the DWG-file directly using Allegro Common Lisp or any other language.

We determined that if we wanted to get CAD data directly from the DWG-file, we would have to use AutoLisp to read the DWG file and write an external file in ASCI format. AutoLisp has only limited string handling facilities so we were unable to implement the transformation to the product model (which requires Lisp Syntax) using AutoLisp alone. We therefore used the AutoLisp-generated ASCI file as an intermediate file and used Allegro Lisp to generate the object-oriented Project-Model from it. All of these transformations currently work in batch mode only.

Having made these strategic design decisions, we designed and developed CIFECAD, our own high-level user-interface to carry out AEC design. In this interface, the user first selects a command from the screen menu such as "PROJECT DATA", "E-DRAWING", "COLUMN", "E-WALL" etc.

CIFECAD then has the following commands for input of non-graphical data:

- The "PROJECT DATA" command is used for input the project information. (*Project code, Project name, Client name, Site location, etc.*)

- The "E-DEF.VALUE" command is used for define the default value of each building element. (*Element size, Element material , etc.)*

The interface opens up dialogue boxes for the user to input the above information. Fig-4.2.1 shows the screen as it appears to a user of CIFECAD.

•We defined building elements using the BLOCK data storage facility of AutoCAD. A BLOCK is a set of graphical entities grouped together into a compound object. We can attach textual information to a BLOCK as an attribute of the grouped graphical entities. For example, we defined a typical Column using the BLOCK facility, whose shape is a 3D-polyline, with attributes such as size, material, and finish code. We did likelwise for other building components such as, Column, Wall, Beam, Exterior wall, etc.

We designed the menu system to capture and extract many of the needed attributes for each element automatically as it is created. A standard CAD system represents only lines and arcs, with no knowledge of what the lines and arcs represent in the artifact being drawn. Using our interface, if we create an element through the beam menu item, the element is automatically labeled as a beam, and much of the detailed drawing is automated. Other attributes of the element which are needed for our planning process e.g., Material, Finish code, or gravity support relationships between elements, are captured by prompting the user for them via a dialogue box in which default values can be predefined. We then write all of these attributes in the BLOCK for that element using AutoCad's built in DWG binary file format.

We next wrote a program using AutoLisp to generate the temporary Database-interface ASCI file from the block information in the DWG file and the file-translation program which gets the data from the temporary ASCI file and creates the PMAP object-oriented project model.

In implementing PMAP, we decided that our model should have the following information in order to describe the building at any stage of AEC process.

- Client information.
- Construction site information.
- Cost information.
- Functional information. (building uses, quality levels, etc.)
- Building elements' geometry and location (Architectural, Structural, Mechanical, Electrical)

- Attribute information for each element (material, color, weight, relationships to other elements, etc).
- Construction duration and budget.

We recognize the desirability of being able to derive the actual model image not only from AutoCad via CIFECAD but also from any other CAD system or application. We propose developing a standard format and content for the temporary file so that we can write system specific interfaces to other CAD systems later to get CAD data from them in the form needed by our object model translator.

In the future, PMAP can be fleshed out so that it not only describes the kinds of information listed above, but can supply needed data to any AEC computer application. There is some question in the researchers' mind about the desirability of creating an overall product model for buildings to support any kind of AEC application, versus defining a series of building models derived from relational database and CAD information known about the project, each aimed at serving a subset of the spectrum of AEC applications. We have not yet approached the limitations of the single integrated solution, even on current hardware, but additional research is needed to answer this question.

Finally, we wrote programs in Allegro Common Lisp to generate input files from a PMAP project model for two separate knowledge-based construction planning systems (OARPLAN and SIPEC). PMAP has all of the information required by these applications and it is easy to extract the required data from the PMAP Project-Model using demons or message passing, for these planning systems. In the same manner, it is easy to generate an object-oriented input file from PMAP for any other application needing PMAP data.

5. Conclusions

Fig-5.1.1 shows a schematic of our current system. We describe the status of its implementation (late summer 1989) in more detail in this section.

5.1 Results to date

1. The user can design a building in CIFECAD using combinations of the following primitive elements. (Fig-5.1.2)
    - **Structural elements:** Column, Exterior-Wall, Interior-Wall, Girder, Beam, Slab.
    - **Architectural elements:** Window, Door, Space(Room) and Floor.
2. The user can define the following attributes for each element.

- Material, Finish code, Size, Position and Angle.

3. The system defines and calculate the following information automatically.(See Fig-5.1.3)

   - Connecting information between elements, area of space and of floor.

4. The system creates the list-expression temporary file from the AutoCAD DWG-file automatically.

5. The user can create a PMAP Project-Model using any of the information defined in this temporary file automatically.(Fig-5.1.4)

6. The system creates correctly formatted object-oriented input files for the SIPEC and OARPLAN knowledge-based construction planning systems.

Although not part of the PMAP system, OARPLAN and SIPEC generate construction plans by reasoning from "first principles" construction knowledge about attributes of the building component data transferred from the CAD model via the PMAP object-oriented building product model. A conventional batch file interface has been developed to output the activity list and sequence logic generated by OARPLAN to MicroPlanner™, a standard Macintosh project planning and scheduling package, which is used to perform network calculations and to display the network diagram graphically. Our integrated design-construction system thus provides a smooth data path to transform building design decisions into project construction networks.

This development of a new way to effect integration of information across project phases is one example of the kind of research being sponsored by Stanford's *Center for Integrated Facility Engineering,* a partnership between Stanford computer science and civil engineering researchers and representatives from all sectors of the AEC industry [Howard 1987].

5.2 Plans for Extension

The present system is a working prototype of a full-fledged object-oriented building product model for information exchange and has been designed to be expandable within its current architecture. We are continuing to develop both the scope of the product model (to permit interfacing it to applications such as cost estimating) as well as adding features to the CIFECAD interface for data entry to the model.

5.2.1 CIFECAD

CIFECAD currently lacks delete, copy and move functions for building elements. We plan to develop these functions. The delete function is probably the most important – and also the most difficult – of these functions for us to add. A "delete" capability is important to permit ease of modification of a design. At the same time, it is difficult to implement because, if the user want to delete an element, the system has to identify and delete all of the appropriate attribute data (including references to the element from other related building elements) along with the graphical data describing the element.

For example, if the user deletes a column then we have to check the wall and girder which are connected to the deleted column. Next we have to check the slab, window, door and space which are supported by, support, are attached to, or include the wall or girder. Finally, we have to check the floor which consists of the wall and space.

In parallel with extensions to the interface, we would like to increase the range of building elements in our system to include, among others, Foundation, Roof, Furniture and other structural, mechanical and electrical elements. Furniture and fittings are important for facility management,a function for which CAD-database systems are being increasingly employed.

5.2.2 PMAP

Our model (PMAP) is by no means a comprehensive model for AEC project data. Therefore we will continue to study means to define and describe the detailed data describing a building project and complex relationships among them in PMAP. The goal of future work on the architecture of PMAP will be to extend it to model not only building elements, but also to incorporate other project attributes, such as soil conditions underlying a project, the neighborhood environment of the construction site, the client's requirements and objectives, and conceptual design data about the project.

5.2.3 Applications which will Link to PMAP

PMAP has been liked to CIFECAD and two Knowledge-based construction planning systems. We have some ideas to extend the generality of PMAP.

The AutoCad BLOCK facility in limited in both the amount and the complexity of attribute data that it can represent. One of our ideas is to link PMAP to a relational database on the Macintosh II (probably ORACLE™) in order to store cost and other ancillary information about building elements. We will then integrate and/or develop a series of knowledge-based decision support systems for functions such as cost estimating, scheduling, time-cost trade-off analysis and building finish design which will use PMAP and ORACLE.

6. References

[Cherneff 1988] Cherneff, J.M.,"Automatic Generation of Construction Schedules from Architectural Drawings," *unpublished S.M. Dissertation*, Department of Civil Engineering, M.I.T., 1988

[Darwiche 1989] Darwiche, Adnan, Raymond E. Levitt and Barbara Hayes-Roth, "OARPLAN: Generating Project Plans in a Blackboard System by Reasoning about Objects, Actions and Resources," *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, (2)3, 1988.

[Geilingh 1988] Gielingh, Win, "General AEC Reference Model(GARM) -- An aid for integration of application specific product definition models," *A PDES publication*, 1988.

[Howard 1987] Howard, H. C., R. E. Levitt, B. C. Paulson, J. G. Pohl and C. B. Tatum, "Computer Integration: Reducing Fragmentation in the AEC Industry," *ASCE Journal of Computing in Civil Engineering*, Vol. 3, No. 1, January 1989, pp. 18-32

[Kartam 1989] Kartam, Nabil A., and Raymond E. Levitt, "Intelligent Planning of Construction Projects with Repeated Cycles of Operation," to appear in the ASCE *Journal of Computing in Civil Engineering, Special Issue: Knowledge-Based Approaches to Planning and Design*, Fall 1989

[Law 1987] Law, K.H., M.K. Jouaneh and D.L. Spooner, "Abstraction Database Concept for Engineering Modeling," *Engineering with Computers*, Vol. 2, pp. 79-94, 1987.

[Law 1989] Law, K.H., T. Barsalou, "Applying a Semantic Structural Model for Engineering Design," *International Conference on Computaers in Engineering*, ASME, 1989.

[Logcher 1989] Logcher, R.Robert, "Better AEC Products: Directions Research at M.I.T.," *CIFE(Center for Integrated Facility Engineering-Stanford University) Symposium Proceedings*, March 1989.

[Pentilla 1989] Penttila, Hannu, "A Scenario for The Development and Implementation of A Building Produce Model Standard," *CIFE(Center for Integrated Facility Engineering-Stanford University) Symposium Proceedings*, March 1989.

[Warthen 1988] Warsen, Barbara, "PDES_A CAD Standard For Data Exchange," *Unix World*, December 1988

[Wiederhold 1986] Wiederhold, G., "Views, Objects, and Databases," *IEEE computer*, Vol. 19, No. 12, pp. 37-44, December 1986.

Linking Knowledge-Based Systems to CAD Design Data with an Object-Oriented Building Product Model[1]

Kenji Ito, Yasumasa Ueno, MASCE, Raymond E. Levitt, MASCE, and Adnan Darwiche

FIGURE CAPTIONS

Fig-3.1.1 HIERARCHY OF PMAP
Our model (PMAP) can be applied to several applications because PMAP models not only building element information but also project descriptiive information such as, building outline, client information, and construction site information.

Fig-4.2.1 CIFECAD INTERFACE
The interface of CIFECAD has various commands for the user to input building elements and project information, to set up floor and elevation information, and to create the temporary file needed to transfer the CAD data to PMAP.

Fig-5.1.1 OVERVIEW OF PMAP
We use AutoLisp to read the DWG file and write an external file in ASCI format. AutoLisp has only limited string handling facilities so we are unable to implement the transformation to the product model (which requires Lisp Syntax) using AutoLisp alone. We therefore use the AutoLisp-generated ASCI file as an intermediate file and use Allegro Common Lisp to generate the object-oriented Project-Model from it.

Fig-5.1.2 EXAMPLE OF DRAWING AND MENU
We designed the menu system to capture and extract many of the needed attributes for each element automatically as it is created. Using our interface, if we create an element through the beam menu item, the

1 This research was funded by seed research grants from the *Center for Integrated Facility Engineering* , Stanford University, the National Science Foundation Grant # MSM-8716608, and the US Army Construction Engineering Research Laboratory. The authors are indebted to these organizations for their support of the research.

element is automatically labeled as a beam, and much of the detailed drawing is automated. Other attributes of the element which are needed for our planning process e.g., material, finish code, or gravity support relationships between elements, are captured automatically, or by prompting the user for them via a dialog box in which default values can be predefined.

Fig-5.1.3 RELATIONSHIPS BETWEEN ELEMENTS

In CIFECAD, when the user inputs a building element, the system recognizes the *connected-to* relationship between elements and store these into the database together with the graphical data. In this example, Girder-08 will automatically be recorded as being connected to Column-14 and Column-15 as it is inserted between them. The connected-to relationship will simultaneously be placed into the descriptions of the two columns.

Fig-5.1.4 PART OF A PMAP MODEL

In CIFECAD, we use only one relationship, connected_to, between building elements, but in PMAP, we recognize other geometrical relation between elements and use the 4 relationships, supported_by, connected_to, attached_on and consist_of in order to express relationships between component instances. Thus: Girder or Beam is supported_by column; Exterior-Wall and Interior-Wall is connected_with Column; Slab is supported_by Girder or Beam; Window or Door is attached_on Exterior-Wall or Interior-Wall; Space(room) consists_of Exterior-Wall and/or Interior-Wall; and Floor consists_of Space.

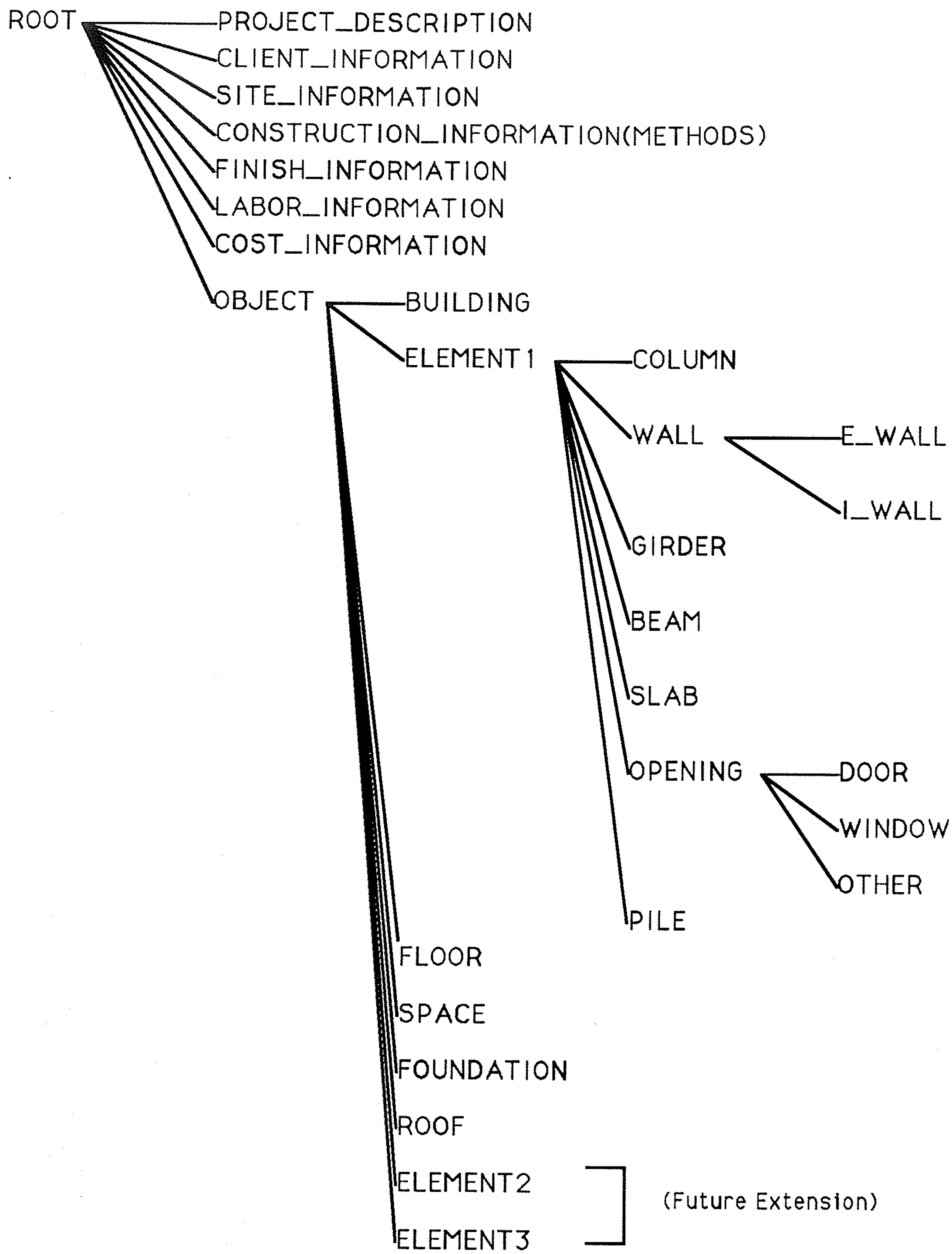

Fig-3.1.1 Hierarchy of PMAP

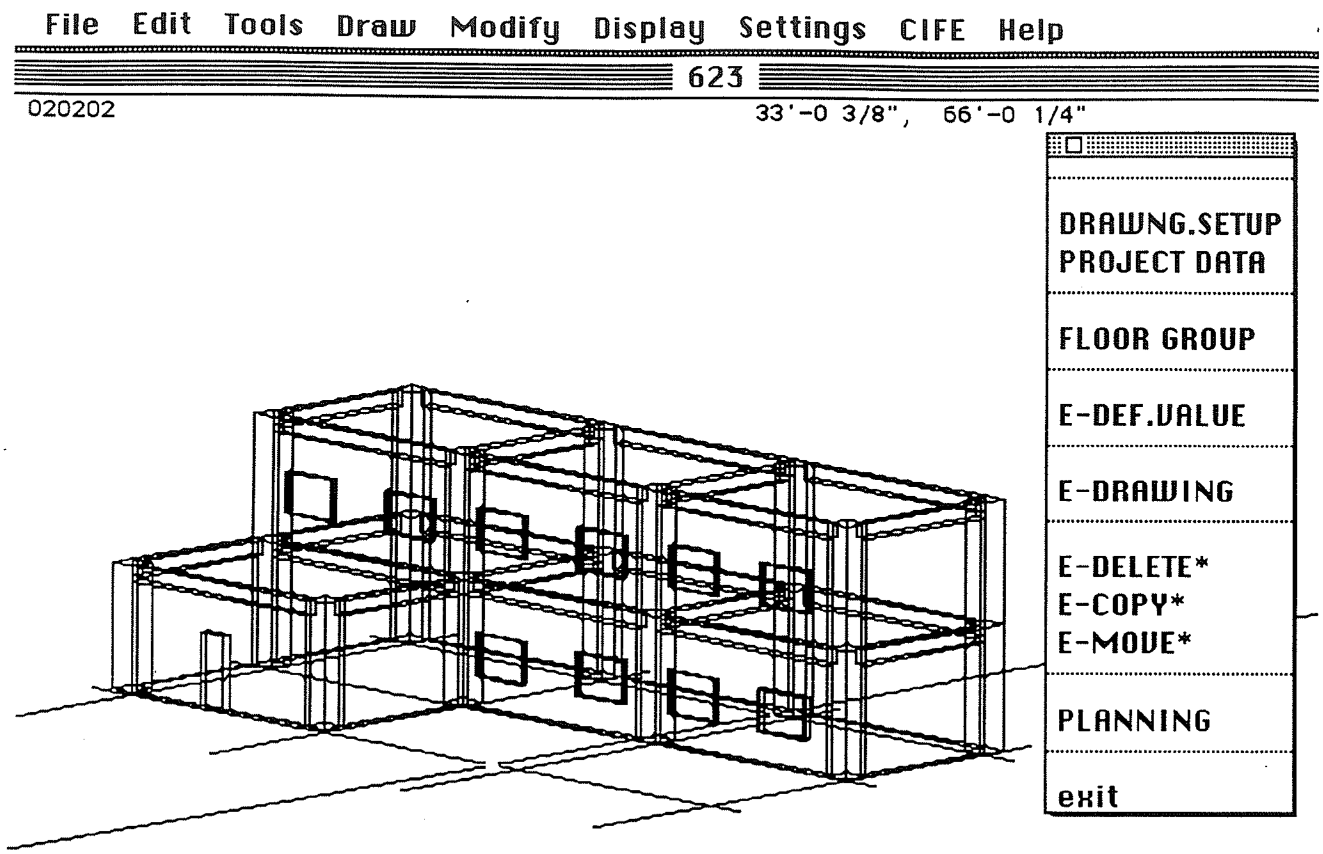

Fig-4.2.1 CIFECAD INTERFACE

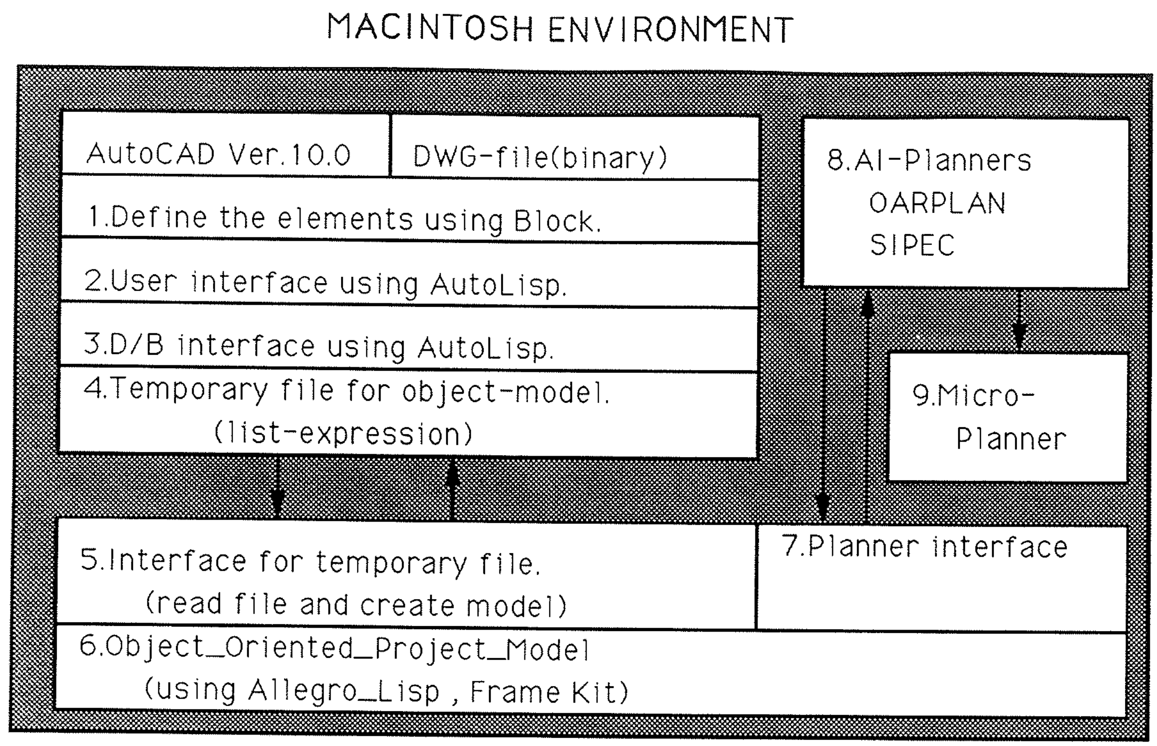

Fig-5.1.1 OVERVIEW OF PMAP

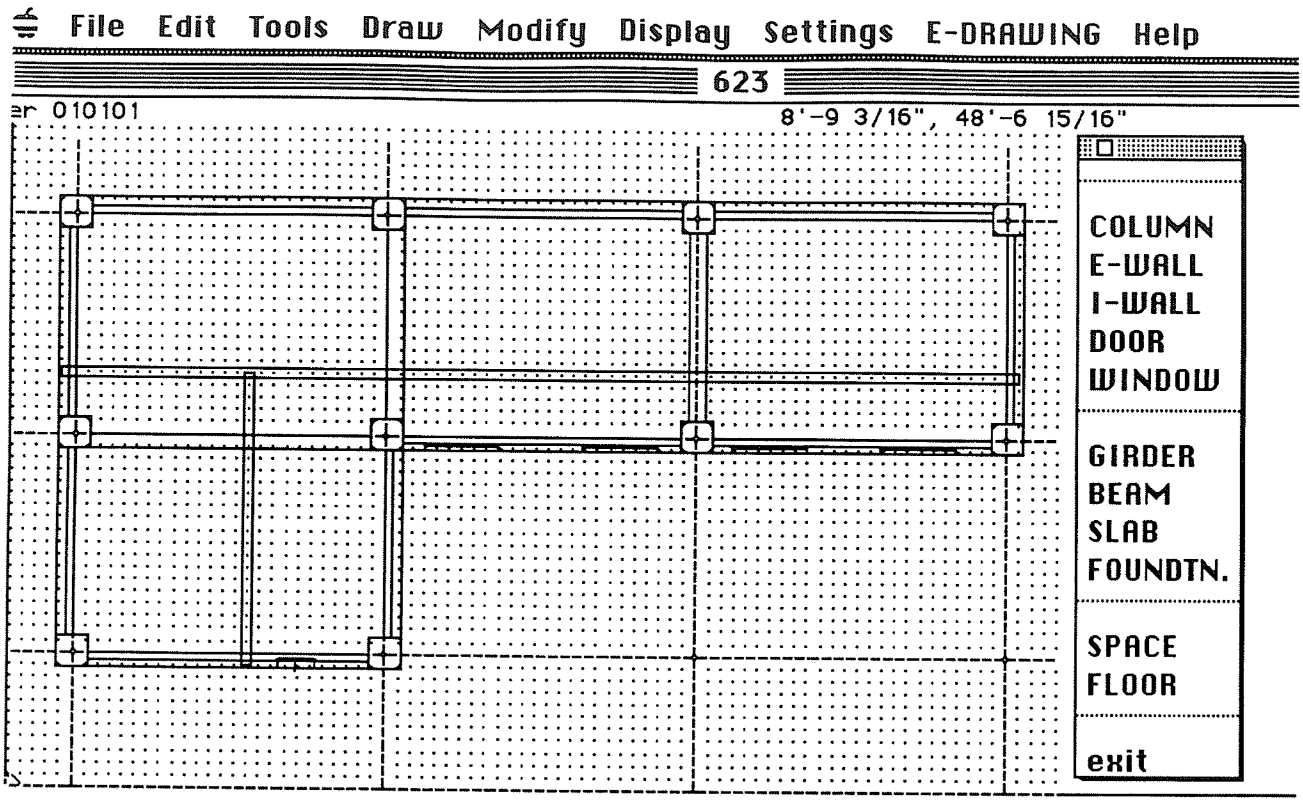

Fig-5.1.2 EXAMPLE OF DRAWING AND MENU

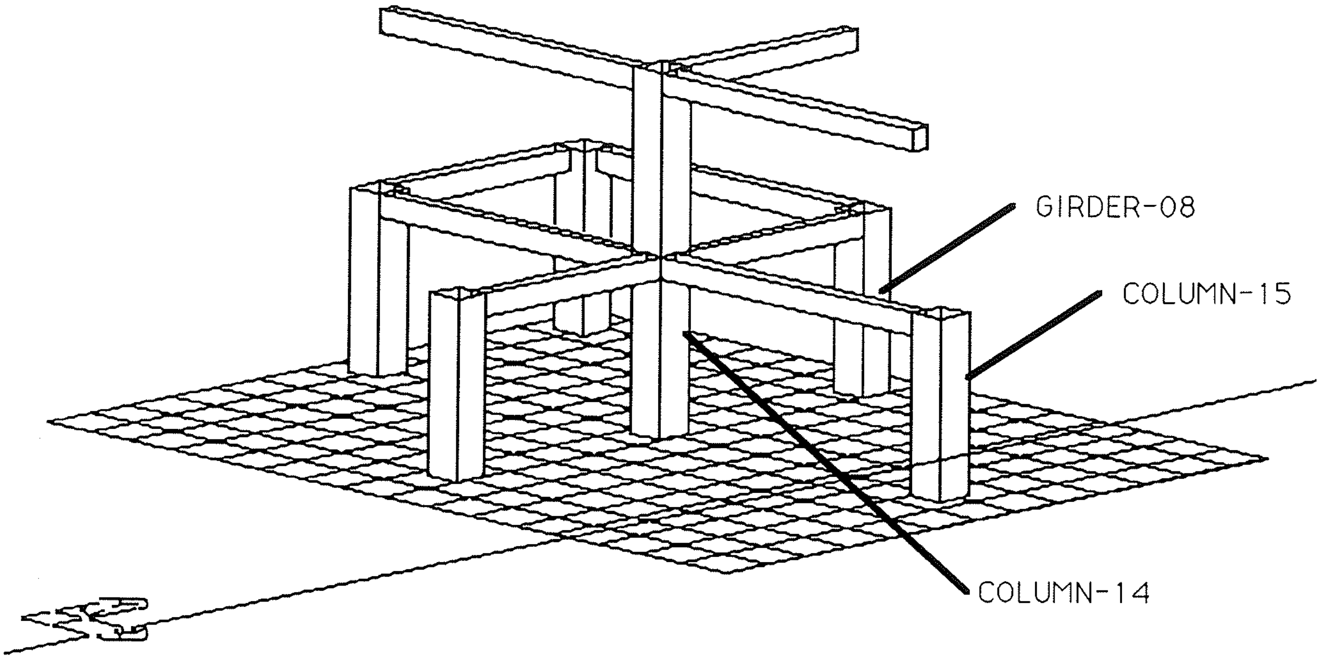

Fig-5.1.3 RELATIONSHIPS BETWEEN ELEMENTS

Columns01	
Supported_by	NIL
Floor_Group	(1)
Angle	(0.0)
Position	((210.0 438.0))
Size	((36 36))
Finish_Code	(1)
Material	(RC)
Instance_of	(COLUMNS)
Columns02	
Supported_by	NIL
Floor_Group	(1)
Angle	(0.0)
Position	((384.0 438.0))
Size	((36 36))
Finish_Code	(1)
Material	(RC)
Instance_of	(COLUMNS)
E_Walls01	
Connected_to	(Columns01 Columns02)
Floor_Group	(1)
Angle	(0.0)
Position	((210.0 438.0) (384.0 438.0))
Size	(8)
Finish_Code	(2)
Material	(RC)
Instance_of	(E_Walls)
Windows01	
Attached_on	(E_Walls01)
Floor_Group	(1)
Level	(36.0)
Position	((300.0 438.0))
Size	((72.0 36.0))
Tyep_id	(1)
Instance_of	(Windows)
Girders01	
Supported_by	(Columns01 Columns02)
Floor_Group	(1)
Angle	(0.0)
Position	((210.0 438.0) (384.0 438.0))
Size	((18 14))
Material	(RC)
Instance_of	(GIRDERS)

Fig-5.1.4 PART OF A PMAP MODEL