

HEURISTICS FOR THE KNOWLEDGE BASE OF A CAD SYSTEM IN ARCHITECTURE.

Hanrot S. GAMSAU (ura CNRS) and GRTC (CNRS)
Ecole d'architecture 70 route Léon Lachamp Marseille FRANCE.

KEY WORDS:

AI, heuristic, CAD, architecture, project, object and logical modelling.

ABSTRACT:

We have developed a knowledge base for a CAD system in architecture using an object oriented environment associated to Prolog. The aim of this research was to make the system a true assistant for the architect. The mock-up we have done shows that this approach is possible if the user is able to control easily the reaction of the knowledge base. Our answer to this problem came from a cognitive analysis of architectural knowledge. It is contained in two architectural heuristics: the working modes and the knowledge masks.

INTRODUCTION:

CAD systems today are nice geometric tools, but they don't know anything about architecture and the architectural project. How can they be made intelligent and able to collaborate with architects during the design process? The team we belong to works on such a system (1). Its mockup is made of three modules implemented with object oriented languages: a geometric modeler, a graphic and symbolic interface and a knowledge base. We concentrate on the knowledge base.

The subject of our communication concerns the heuristics that this base requires to organize its reasoning.

Since architectural knowledge is vast and complex, it is not possible to deal with all the knowledge in a given computer aided design situation. Doing that would result in unpleasant user and machine interaction: numerous questions and delayed answers.

Our cognitive analysis of architectural knowledge has shown that, in a specific stage of the project, architects implicitly isolate the relevant knowledge they need for designing.

For such a selection, they use two heuristics. The first one allows selecting knowledge on the basis of different semantic points of view and of the definitions of architectural object at different levels. The second one gives access to knowledge depending on the current type of operation.

In a previous communication [Hanrot,1988] we have shown that the first heuristic may be modeled with filters. Here we focus on the second one.

First, we will introduce an example for which our base will use this heuristic. Using this example as a guide, we will then answer the following two questions:

-**Why** does the knowledge base behave in such a way?

-**How** does it work inside the system?

The first answer will be proposed from a cognitive point of view referring to the "problem solving" approach and to theories of architectural project. The second one is relevant to computer science and will deal with object representation based on frames (2) associated with predicate logic (3).

2-Knowledge-base functionalities.

The base extends the domain of knowledge that CAD systems generally know. So, it knows how to make up decomposition of a building in a set of parts, it can trigger architectural rules concerning alignment, composition. But, more than this description of architectural objects, the base knows the nature of documents and conditions of object representation, i.e. scale, scope and extent of work, representational conventions. It also knows architect himself and his assistants, i.e. draftsmen, quantity-surveyors. These actors are qualified with skills and action capacities.

A blackboard:

Management of this extended knowledge requires a new element in the architect's working environment. It is a blackboard in which all of base's control parameters during the project are grouped.

This blackboard is made of three sections:

1-WORKING-MODES	2-MASKS	3-STATE
PROGRAMMING	DEFINITION LEVELS	CONTEXTUAL
DESIGN	VIEW POINTS	PROJECT
EVALUATION	KNOWLEDGE ORIGIN	

Fig-1-Black-board

The **first** one contains parameters concerning working modes with which the architect can work, i.e. programming, design, valuation. Each mode implies a specific behaviour of the knowledge base. The use of these modes is not linear, the architect can swap from one to the other at any time.

The **second** one is made of different knowledge mask, i.e. definition levels, view points, knowledge origins. These masks allows the architect to make activate or deactivate different fields of the knowledge base. In the mock up we have done, the masks can take the following value:

-Definition levels: {entities, divisions, elements, constituents}

-View-points: {morphological, economical, usage}

-Origin: { doctrine, rules of the art}

The **third** one concerns two possible states that architectural objects and documents can take. If they are declared "contextual", they are not modifiable and any kind of solution which needs their transformation is not acceptable. On the opposite, objects or documents declared "project" can be transformed.

These different sections are interdependent. According to the working mode selected, the mask and state definition changes. But an example of manipulation will help us explain how knowledge is requested.

An example of manipulation:

It will concern Brigitte an architect who is working at Caro Architects. The project concerns an urban residential building and she works on preliminary working drawings.

A-The goal of **programming** is, on one hand, to describe the existing context: we call it "description". On the other hand, it is to declare requirements on objects that we have to design: we call it "prescription".

A1-Description is the elicitation of constraints that existing objects (building, garden, street...) impose on the object to be designed. But, at this time, the base does not carry out this complex function. As there is no controlling, all the knowledge is masked and state-3 is fixed as "contextual". Let us follow the dialogue between Archibase (that is the name we gave the base) and Brigitte.

Nature of blackboard:

1-Working-mode: {description}

2-Masks:level {nil}-View-point {nil}-Origin{nil}

3-State: {as context}.

1-**Brigitte**: I describe contiguous building: building1!

2-**Archibase**: (Archibase create building1)

3-**B**: Building1 has a squared shape.

4-**A**: (Archibase gives this property to building1)

5-**B**: ...

A2-Prescription consists in the declaration of requirements for the object we have to design. A part of these demands comes from the programme that the client gives the architect. But it is often insufficient, and architects have to complete it as project goes on. Also, information state is as "project", and, by default, all knowledge is active:

Nature of blackboard:

1-Working-mode: {prescription}

2Masks:level{entity,division,element,constituant}viewpoint{morphological,use,economical}
Origin {rules of the art, doctrine}

3-State: {project}.

6-**Brigitte**: All the openings will be made of PVC!

7-**Archibase**: (Archibase declares this constraint on the openings' definition)

B-The aim of **design** is to propose an architectural solution which agrees with the constraints or requirements known by the base. It can be drive "step by step" or by "models".

B1-Driven by models, the base is "active". It helps the architect to find architectural models convenient to his problem. When a choice is taken, the base helps the architect to define the visible properties and to create links with other objects.

Let us consider that a residential building, building2 made of three floors has been designed. Now the goal of Brigitte is to design an apartment on ground floor. According to french regulation there could be such a dialogue:

State of black board:

1-Mode: {design by models}

2-Masks:levels{division,element}-viewpoint{morphological,use}-Origin{rulesoftheart, doctrine}

3-State: {as project}.

9-**A**: what do you want to design: Division: floor, appartement ,room, stair,
Element: wall, bay, floor, roof,?

10-**B**: appartement!

11-**A**: what type: 1-room, 2-room, 3-room, 4-room, 5-room, 6-room?

12-**B**: 3-room

13-**A**: I am working (Archibase creates 3-room-1 in the project base)

What are its functions: residential, work ?

14-B: residential!

At this time Archibase proposes to Brigitte all the visible properties of building1, and waits for a value. Then it tries to define the relations to which building1 could belong to. If the other objects don't exist in the project base, Archibase proposes to create them:

15-A: To what floor 3-room-1 belongs to: Ground-floor, first, second?

16-B: Ground-floor!

17-A: May I create the rooms of 3-room-1, yes/no ?

18-B: yes!

19-A: I am working! (archibase creates a corridor, an entrance, two sleeping rooms, a bathroom, a WC, a closet, a living-room, a kitchen)

20-A: As it is an apartment on ground floor, corridor must have a width not less than 90 cm, because physically handicapped must have an access, yes/no?

21-B: yes!

B2-With step by step mode, the base is only reactive. To manipulate the base , Brigitte can trigger actions using a menu. Let us take the previous example with this mode.

State of black board:

1-Mode: (design step by step)

2-Masks:levels(divisions,elements)-viewpoint(morphological, use)-Origin (know-how, doctrine)

3-State: (as project).

22-B: I design an apartment!

23-B: It is Type3: Type3-1

24-B: Give me the list of its parts?

25-A: a corridor, an entrance, two sleeping rooms, a bathroom, a WC, a closet, a living-room, a kitchen.

26-B: I design a corridor: corridor1!

27-A: Of what floor type3-1 belongs to: Ground-floor, first, second?

28-B: Ground-floor!

29-A: As it is an apartment on Ground-floor, corridor must have a width not less than 90 cm, because physicaly handicapped must have an acces, yes/no?

30-B: yes!

C-The aim of valuation is to give an expertise and some times to modify an architectural solution completely or partially designed. This valuation is relative to one view-point and to well defined definition levels.

We can note that the economical view point has always been masked in the previous examples. Let us suppose that Brigitte wants now to have a valuation from this view point.

State of black board:

1-Mode: (valuation)

2-Masks: levels (divisions, elements) -view point (economical)-Origin(Know-how, doctrine)

3-State: (as project).

31-A: What financial help do you expect for the project: PLA, PAP, None? (4)

32-B: PLA!

33-A: As it is PLA, it would be nice to have a wash basin in the WC, because that could raise fiancing, Yes/no ?

34-B: yes!

These examples show us that knowledge-base functionalities are different according to the working mode chosen by the architect. Masks of knowledge are associated to these modes and allow the architect to implicitly specify the

knowledge domain which is pertinent according to his architectural problem. Let us try to define the reasons of these functionalities.

3-WHY DOES THE KNOWLEDGE BASE BEHAVE IN SUCH WAY?

Before the base development, we had to analyse knowledge on architecture and project. This analyse has two bases. One is an introspective effort of our own practice as an architect. The other is a more theoretical frame made up of architectural theories on architecture and project [Boudon,1977; Perouse, 1972; Choay,1980]; reflexions about CAD in architecture [Quintrand, 1985; Giraud,1985, Alexander, ; Negro Ponte,]; progress of cognitive psychology and Artificial Intelligence on problem solving and design [Minsky,1981; Simon,1983; Akins,1978].

From this approach, we elicited a corpus of concepts which expresses architectural knowledge. The most important concepts are the "types or models" and the "rules" that constitute the basic elements of knowledge, and also the "organization", the "exploitation" and the "elaboration" of these elements.

A-Types/models include: Participants of the project, the documents, the manipulated objects and the relations they maintain. In our example, these different types can be illustrated like this:

```
-participants: {architecte{Brigitte}}
-document: { folder{detailed-project}}
-manipulated-object: {entity{building},division{apartment,floor, room.}
-relation: { Inclusion, aggregation/partition}
```

Types/and models are qualified with properties.

B-Architectural rules can be classified in three families according to their origin. Rules of the art, methodological rules and doctrinal rules. In our example, we triggered two rules, the first one is a Rule of the art coming from regulation, the second one is coming from doctrine of Caro Architects:

```
Rule 1:      If an apartment is on ground floor ,
              Then corridor must have a 90 cm width,
              Because a physically handicaped must have an access.
Rule 2:      If financing of an apartment is PLIA,
              Then WC must contain a wash basin
              Because that could indreases financing.
```

C-There are four organizations of knowledge. First of all there are two taxonomies by "generalization/specialization" and by "aggregation / partition". These classification modes exist in a lot of different domains [Minsky,1981]. The "definition levels" and the "view point" seem more specific to design processes and specially to the architectural one.

Definition levels are characterized by a correspondence between the partition taxonomy of the architectural object, geometric scales of documents and project phases.

ARCHITECTURAL OBJECT	SCALE	PROJECT PHASE
Global {park, lotissement...}	1/1000,1/500	Urban project/ Design rresentation
Entity {building,hospital...}	1/500,1/200	Design rresentation/ permit drawings
Division {floor, room, structure...}	1/200, 1/100	permit drawings
Element {Wall, opening, entablature..}	1/100, 1/50	permit drawings/ construction drawings
Constituant {brick, extrusions...}	1/50,1/20...	construction drawings

Fig2 -Definition levels

View points result from a correspondence between the architect skills, architectural view and document types.

SKILL	ARCHITECTURAL VUES	DOCUMENT TYPE
Construction	Ouvrages	Descriptif...
Implementation	Oeuvres	Clauses Techniques Particulière...
Economical	Cost	Construction cost, Financial...
Geo-physical	Organum	Technical folders...
Urban	Ubain element	Building autorization...
Use	Function	Programme, spatial allocation ...
Morphological	Shape	Graphic represent., Descriptif...

Fig3-View-points

According to their arguments, architectural rules can be qualified with specific definition levels and view points. So, rule1 has definition level:{division (because corridor is a division)} and as view point: {morphological, use}. Rule2 has definition levels: {division and element (because wc is a division and wash basin is an element)} and as view point: {economical}.

D-Two types of heuristic characterize knowledge's exploitation: some of them are mostly explicit, the others are implicit because no explanation is given.The first group concerns the different actions that architects develop (i.e. creation, renovation...), the working mode, the project methods and strategies and their rules. The second group concerns "masking" of knowledge that the architect naturally does by different representation mode. This masking use the view points and definition levels that we have seen previously.

We have seen from where the architectural heuristics come, we now have to see now how we have transferred knowledge to the base.

4-HOW DOES IT WORK INSIDE THE SYSTEM?

Modeling's aim is to express architectural knowledge according to a specific conceptual model. In our case, it is an object oriented model based on frame and associated with predicate logic. For implementation, we use programming languages close to the model: Objlog (2) which is an object layer over PrologII.We can resume knowledge transfer with the following board:

Architectural corpus	->	Objlog / Prolog
Types / models	->	Class and Instance
Properties	->	slots and facets, and multiple inheritance
Relations	->	Reference slots
Rules of know how of doctrine	->	Domains controls and Daemons, Methods / Prolog Clauses
Organizations		
Taxonomies	->	Vertical/selective multiples inheritance
Definition levels	->	Classes and slots
view-points	->	Classes slots and multiple inheritance.
Exploitations		
Actions	->	Methods and Prolog programs
Working modes	->	Methods and Prolog programs
Masking	->	Filters with programmes Prolog programs
States	->	Slots and Prolog programs

Fig4-Knowledge transfer.

The following figure illustrates the graph of objects we dealt with in our example. Let us note that Archibase is made up of two parts: The knowledge base itself and the project base. The knowledge base is true for a lot of project, and it is a graph of classes linked with vertical inheritance, which express specialization ("kind-of" link). Other links are specific of architectural domain. Project base is a graph of instances true for a particular project. Instances are linked to their classes with a "is-a" link, to which is associated vertical inheritance.

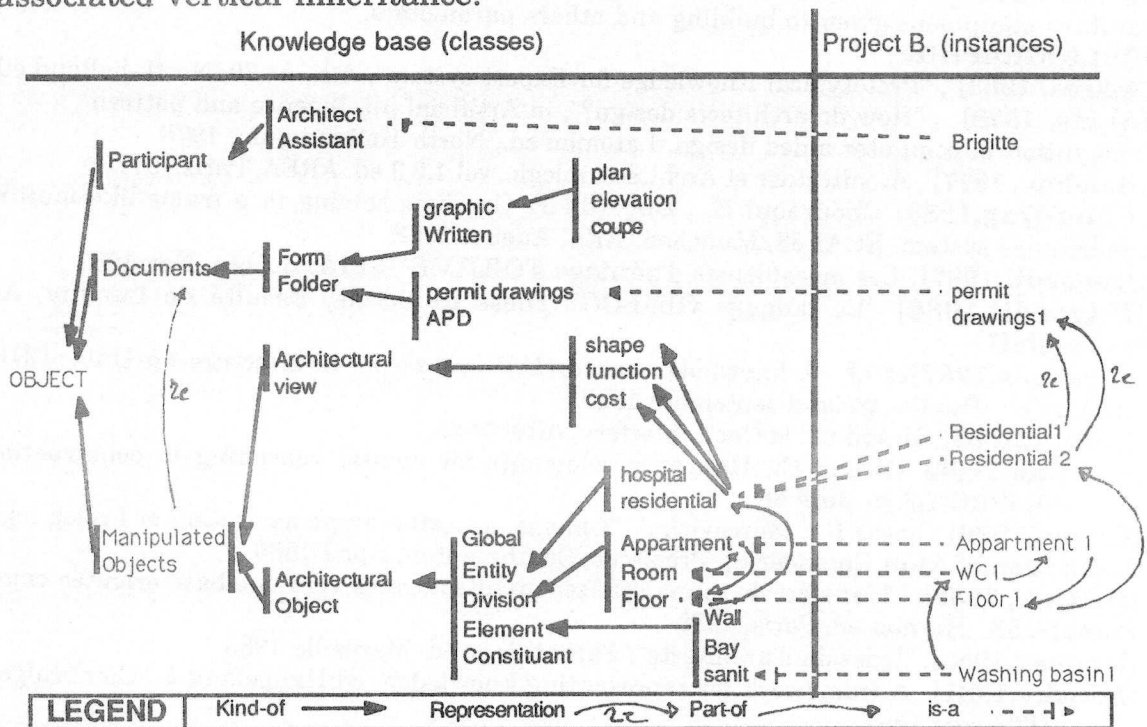


Fig 5-Knowledge base graph

5-CONCLUSION

We presented in this paper a knowledge base which is not only an expertise tool on a finished architectural object, but also an interactive assistant for

the architect. To do this, we analysed architectural knowledge and we transferred two kind of heuristics used by architects during projects. Some are explicit as "working modes" others are implicit as "knowledge mask". But even with these original functionalities, the base is not perfect. From a cognitive point of view it would be necessary to have a more precise modeling about knowledge that architects exploit during project, then the base could assist architects in project methodology.

From a computing point of view the basic languages we use must be faster, and we think that filtering mechanisms that we proposed for knowledge masks could be optimized and integrated in the object language (Objlog).

Last, a large work must be done to extend the knowledge base and to make it capable of functioning in the real conditions of projects in an architectural office.

NOTES:

(1) System's name is Tecton and the team is managed by Mr Quintrand (GAMSAU) and Mr Chouraqui (GRTC). This research is sponsored by the French ministry for building: MELATT.

(2) Objlog is an object layer over Prolog developed at the GRTC laboratory (Marseille).

(3) Prolog is PrologII of Marseille.

(4) PAP and PLA are financial aids for public housing. They are computed according to the sanitary equipment given to building and others parameters.

BIBLIOGRAPHIE:

[Aikins, 1983], "Prototypical Knowledge for Expert system", A.I., n° 20, North-holland ed.

[Akins, 1978], "How do architects design?", in Artificial intelligence and pattern recognition in computer aided design, Latombe ed., North Holland pub., 1978

[Boudon, 1977], Architecture et Architecturologie, vol 1,2,3 ed. AREA, Paris,1977

[Chou-Dug,1988] Chouraqui E., Dugerdil P. Conflicts solving in a frame-like multiple inheritance system. ECAI 88, Munchen, RFA, August 1988.

[Dugerdil,1987], "Les mécanismes d'héritage d'OBJLOG" RFIA, Antibes, Nov.1987

[Dugerdil,1988] "Le langage OBJLOG" Thèse de 3cycle, Faculté de Luminy, Aix-MarseilleII.

[François,1987]et al. -A knowledge based CAD system in architecture on Unix, EUUG Autumn'87, Dublin, Ireland septembre 1987

[Giraud,1985] Giraud C, "la Cao", interface, Afcet 1985

[Gir-Han,1988] Giraud C., Hanrot S."Elements for spatial reasoning in construction", Proc. 5th ISRC Tokyo Juin 88

[Gloess,1986] Gloess P., Marcovich J, "Oblogis, a flexible implementation of Prolog logic", Application of AI in Engineering Problems, Southampton, April 1986

[Hanrot, 1988], "Transfert de connaissances architecturales vers une base orientée objet", EuropIA 88, Hermes ed. Paris, 1988

[lebahar,1983], "ledessin d'architecte", Parenthèses ed. Marseille 1983

[Minsky,1981] A framework for representing knowledge. ed.Haugeland J., Cambridge, MA: MIT press.1981

[Negroponte,1975], "Computer aids to design and architecture", Van nos reinhold ed. 1975

[Perouse,1972] - Le vocabulaire de l'architecte, ministère des affaires culturelles- Paris, 72

[Quintrand,1985] Quintrand P. et al - La CAO en architecture Hermes édition, 1985.

[Simon, 1983], "search and reasoning in problem solving", A .I., n°21-1-2, North-Holland

[Tecton, 1986] Chouraqui E., Quintrand P. et al. - "Le projet TECTON ". CAO robotique en architecture et BTP, Marseille