DEDALE :

AN EXPERT SYSTEM FOR THE CONSTRUCTION AND THE RESOLUTION OF MULTICRITERIA DWELLING DESIGN PROBLEMS

P. LE GAUFFRE - M. MIRAMOND

INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON
Département Génie Civil et Urbanisme
Laboratoire Méthodes - Bâtiment 304
20 avenue Albert Einstein
69621 VILLEURBANNE CEDEX
FRANCE

SUMMARY

The works presented in this paper concern the use of a rule-based system, furtherly interfaced with the CAD system "X2A" [1], to provide an aid to the designer in building and solving multicriteria problems.

Since the building quality is evaluated with several points of view (architectural, thermal, acoustical, economical ...), we can consider the problem of dwelling design as a multicriteria decision making problem.

So instead of considering dwelling design as an iterative process of generating and evaluating alternative solutions until a satisfying one is found, we propose an iterative process of generating and solving restricted multicriteria problems.

We present in this paper a prototype of an expert system, not yet definitive, dedicated to the four tasks :

1°-modelisation of the dwelling project

2°-selection of a strategy for modifications

3°-construction of a restricted multicriteria problem (according to the strategy)

4°-management of the resolution process.

7. CONCLUSION

Even not finished, our expert system prototypes suited human experts, because they have seen early what a reasoning may look like.

We can already say, that the success of a complete expert system depends on distribution and maintenance conditions at both software and knowledge levels.

CSTB has software competences, and many experts in every building fields. It is a public utility with professional building contacts. Therefore CSTB is settled at a privileged crossroad to analyze different building fields to find those which may benefit from an expert system approach, or any other artificial intelligence tools.

To control the real execution is a little different because there is no more paper with all information corresponding to one step. But at one moment during the last step there is only partial information relevant of the waterproofing work state. As the two previous steps, the inspector (and so far the system) has to check many details and finally produce a report.

For the control of the elaboration on the building field, the system do not work any more in interactive way. The expert system gives the inspector a list which describe every things is got to find on the flat roof, and some brief lists concerning the particular stage of the elaboration he is going to control. Those lists annotated with observations got on the building field, will then serve to enter new information in the system, which will at least furnish the remarks to write the final report.

In fact, a quite long time - few monthes - passes by between each step. When studying execution documents, many information entered at the previous step must be kept and some others need to be confirmed before. At the elaboration step, previous information are not questioned, they are printed to be checked on the roof flat. So far the facts base is empty when the system starts the study of the design documents, and then filled up by them. Next step it grows up by adding execution documents information. Final step, a big part of the facts base is the first system output, then it is completed by facts seen on the building field.

As we sais before, this treatment needs a forward chaining inference engine. This engine has been developed for this application. It replaces the backward chaining inference engine of our expert system shell ELSE, but we kept its interfaces. This last application is going to be transfered on Macintosh.

6. TECHNICAL CONTROL IN WATER-TIGHTNESS DESIGN

This last expert system prototype simulates technical inspector reasoning on the well defined subject : waterproofing work on flat roofs. One of the main function of the system is to help the inspector to make a complete check, avoiding omission.

The work of a technical inspector consist of three steps :

- designing report
- execution documents
- elaboration in the construction field.

To end each step the inspector lays down results of his verifications in a report. The system had to help him to produce those reports.

The controller, during the two first steps, works on paper supports. Each support contains the information needed to evaluate it. Make a technical control consists at each step to write in a report the result of the analysis of the studied paper.

This analysis distinguishes three types of formulations :

- "Prescriptions" which impose modifications because of the regulation,
- "Recommendations" propose well-know solutions, and give important warnings,
- "Advices" are expert suggestions, for example : a better technical solution on a long period, but more expensive.

The reasoning to be done on these two first steps is clearly a forward chaining inference working on a facts base constituted of the studied documents. The rules base contains all the tests and controls that may be done, and depending of test results every remarks (Prescriptions, Recommandations, Advices) that may be written in a technical control report.

and even rules statistics

Fin - Affichage Impression Sauvegarde

fichier-stat

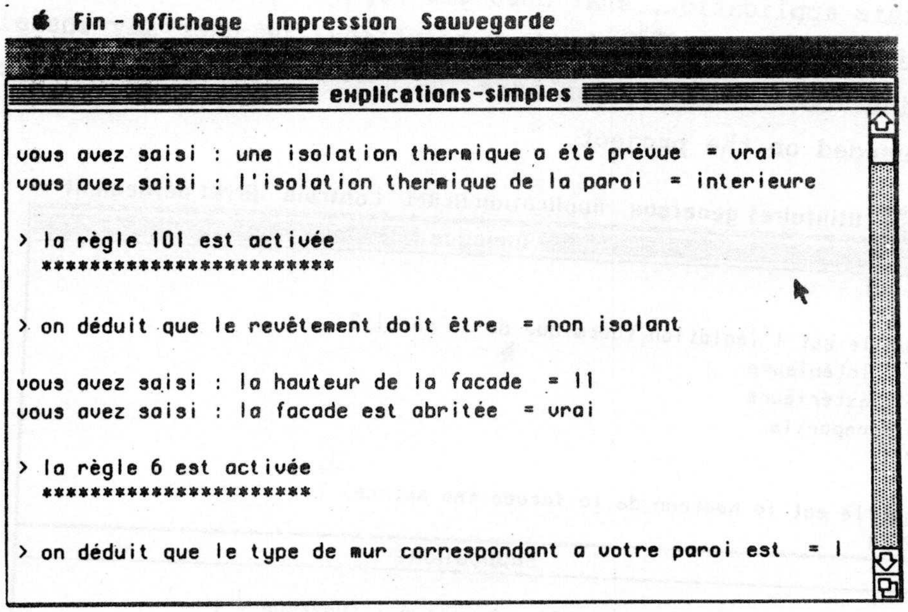
règle n° :	1	2	3	4	5	6	7	8	9	10
utilisée :	11	12	10	12	10	7	9	9	14	9
règle n° :	11	12	13	14	15	16	17	18	19	20
utilisée :	10	11	12	10	4	4	8	8	6	4
règle n° :	21	22	23	24	25	26	27	28	29	30
utilisée :	7	15	7	16	10	14	24	14	24	31

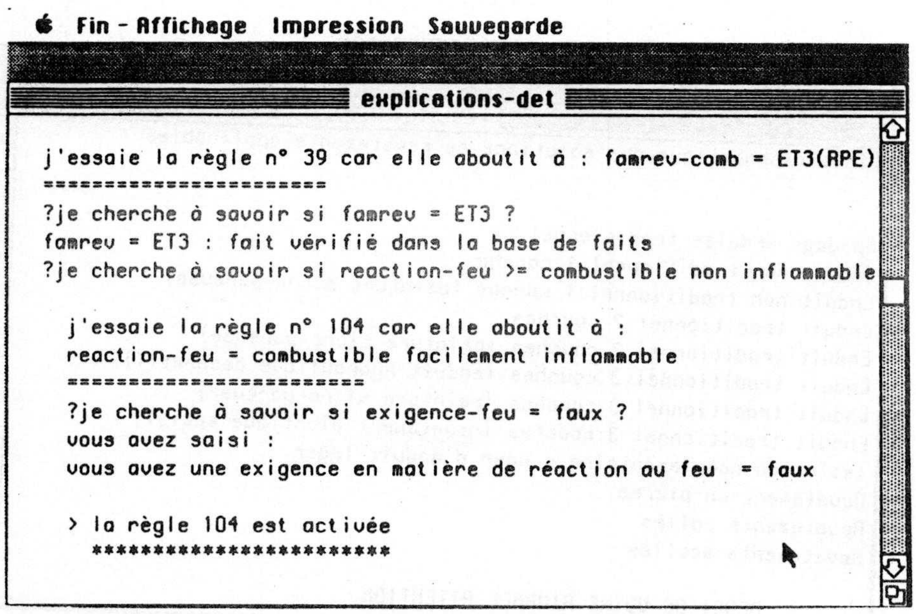
Another interface then allows the user to display rules.

```
 Utilitaires generaux  Application DERM  Controle  Arret défilement
                     Affichage d'une règle
Numero de la regle : ? 14

regle numero 14  utilisee jusqu'alors  10 fois

Si     la hauteur de la facade >= 6
 et    la hauteur de la facade < 18
 et    le revêtement doit être non isolant
 et    la facade est abritée faux
 et    [  la situation de la construction situation a
       ou la situation de la construction situation b
       ou la situation de la construction situation c ]
 et    l'épaisseur de la paroi est <  2.750000E1

Alors  le type de mur correspondant a votre paroi est II

        ..............  Tapez RETURN pour continuer  ..............
```

```
 Utilitaires generaux  Application DERM  Controle  Arret défilement
                     Affichage d'une règle
Numero de la regle : ? 43

regle numero 43  utilisee jusqu'alors  14 fois

Si     le revêtement doit être isolant
 et    le type de mur correspondant a votre paroi est <= XIV
 et    la réaction au feu de classe : >= combustible non inflammable
 et    l'aspect desiré est clin
 et    la résistance aux chocs du revêtement <= moyenne

Alors  une solution de revêtements possible est : VET X

        ..............  Tapez RETURN pour continuer  ..............
```

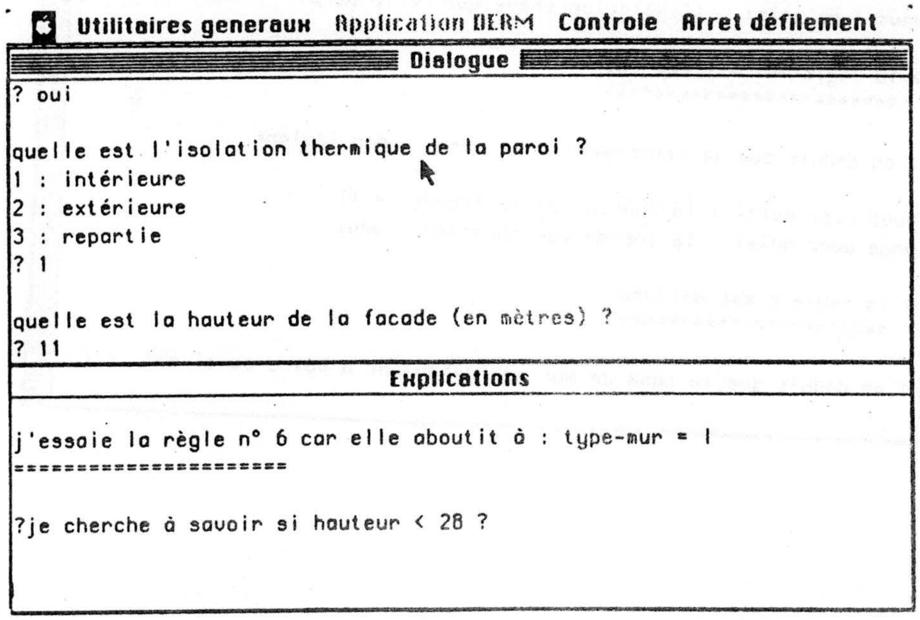
When the expert system has finished, the user is able to consult all information entered and deduced, and also some explanations (detailed or not).

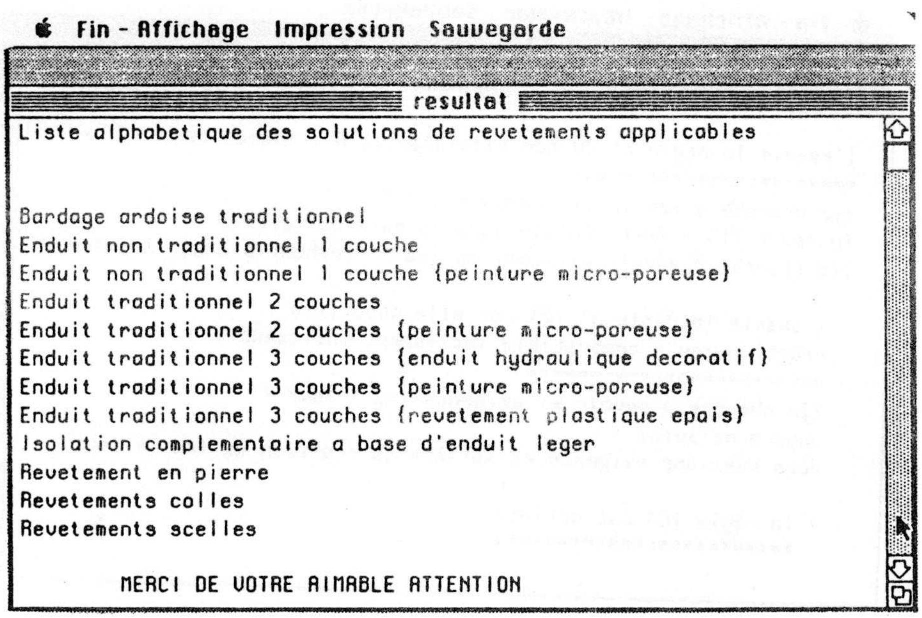




Example :

Let see, on this application, what does the expert system prototype :
The system finds all external cladding solutions the user may install on his building. So far the system asks the user to get the minimum of information needed on the project.





5. AIDED CHOICE OF EXTERNAL CLADDING

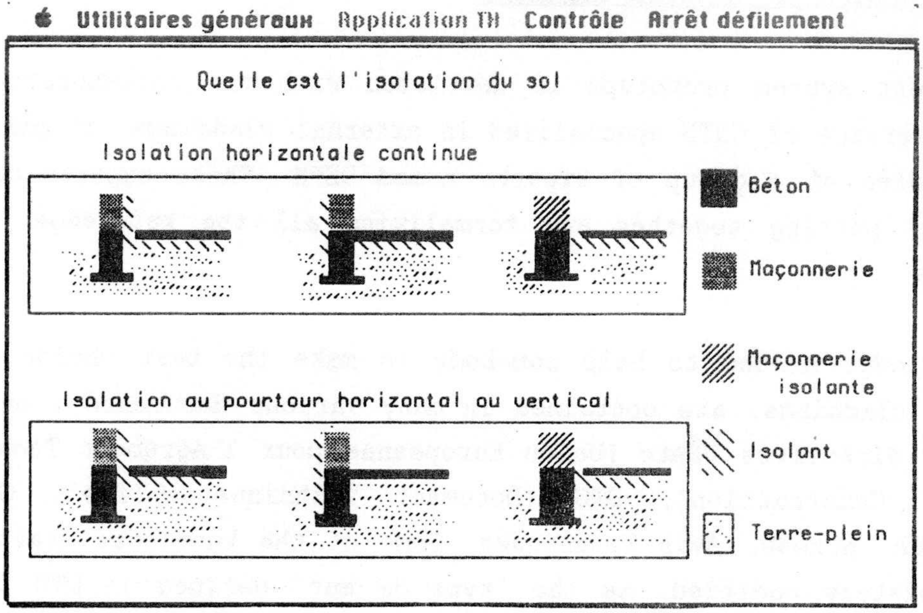
This expert system prototype is developed with the collaboration of another service of CSTB specialized in external claddings, in parallel with studies of a group of experts named DERM. These experts are in charge of putting together and formalizing all the knowledge about claddings.

The knowledge needed to help somebody to make the best choice about external claddings, are contained in many various documents : cahiers du CSTB, directives UEAtc (Union Européenne pour l'Agrément Technique dans la Construction), DTU (Document Technique Unifié), Guides Techniques, normes, Avis Techniques. Some of the knowledge statements are completely codified, as the "type de mur" defined in DTU 20.11. Some others are more qualitative, they can't be appreciated in every situation at the same level, for example : cladding stability, installation difficulties.

The ultimate aim assigned to the expert system prototype is to give the user a sequenced list of external cladding solutions corresponding to specific constraints of a given project.

Before classifying solutions, a first step of selection of accepted solutions is done on imperative constraints, such as wall nature compatibility, water tightness, wall thermal insulation, fire resistance... Then to classify from the best to the worst selected external cladding solutions, the expert system considers more subjective criteria. Therefore it is the user who judges per criterion its degres of importance. Some examples of criteria : architecture adaptability, skilled labour qualification, sensibility to specific climate conditions...

To enter subjective criteria, importance degreas, notes... we have developed a set of specific interfaces which have been generalized enough to be used by other applications which need to classify their results.



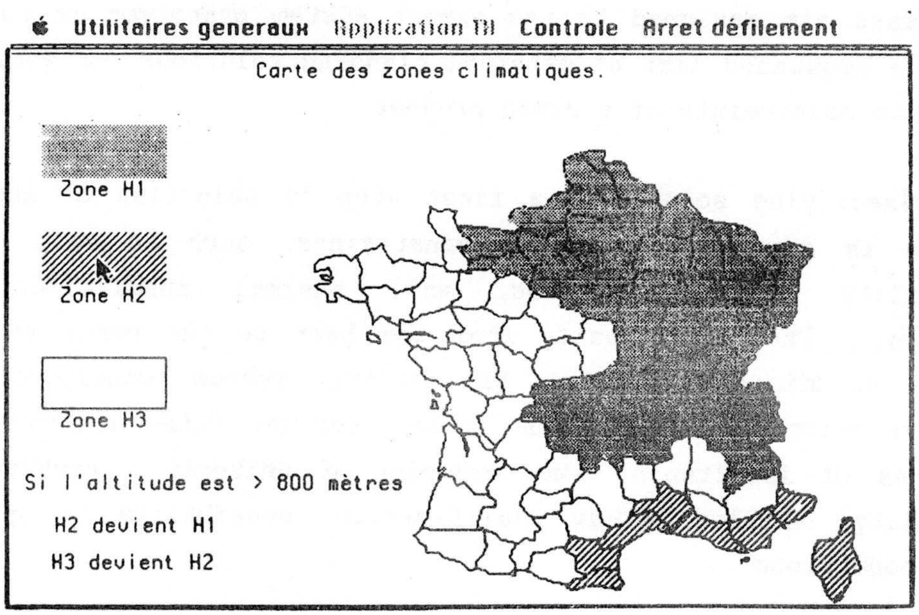


Figure 6. Simple drawings instead of long question.

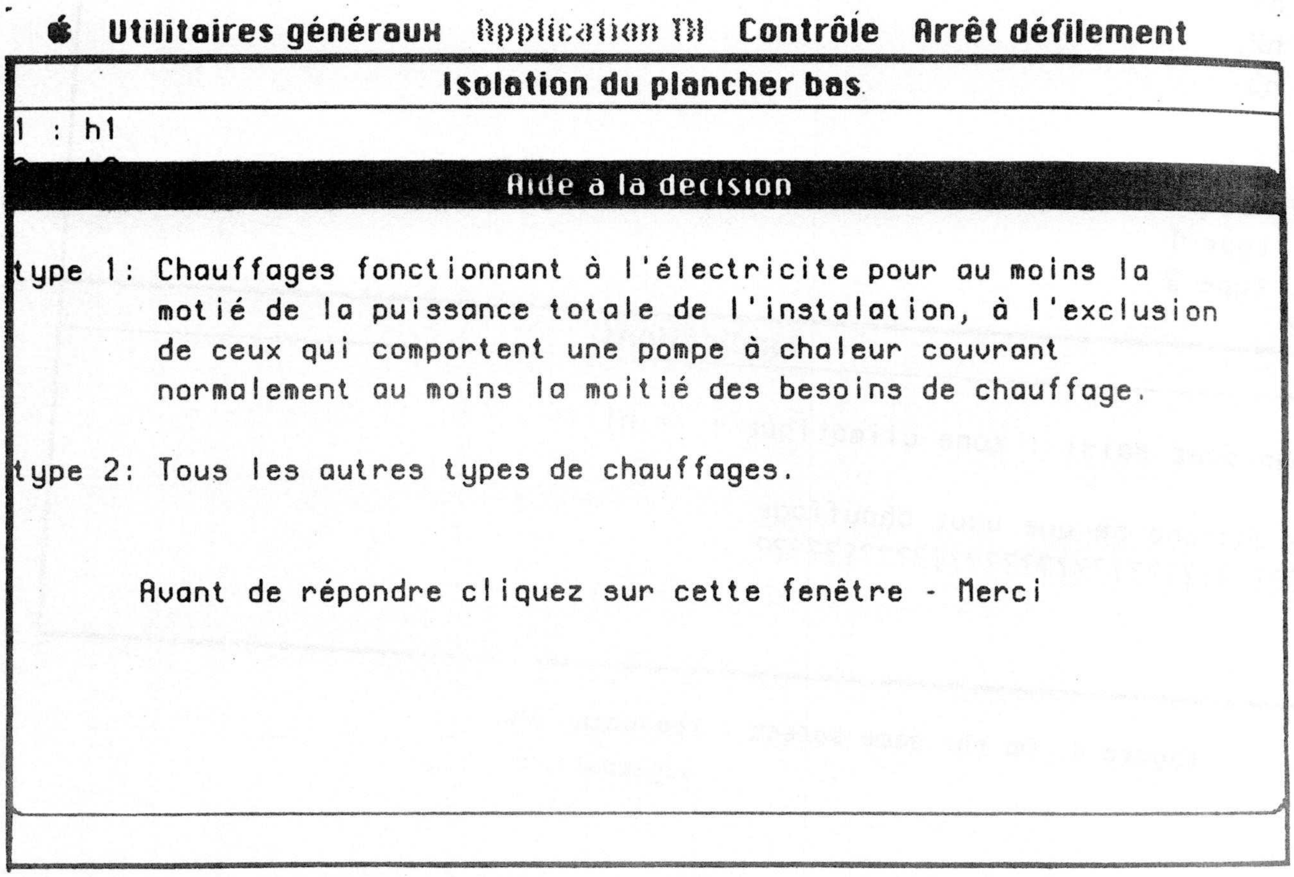


Figure 5. A window to help the user to auswer a question

In parallel, the third part of the project consist in transfering the application running on a VAX 11/750 onto a Macintosh Plus. Now we may use Macintosh characteristics : many windows, one for the dialogue with the user, one for explanations, one per menu... ; many drawings to help the user to answer questions (fig.4, 5, 6).

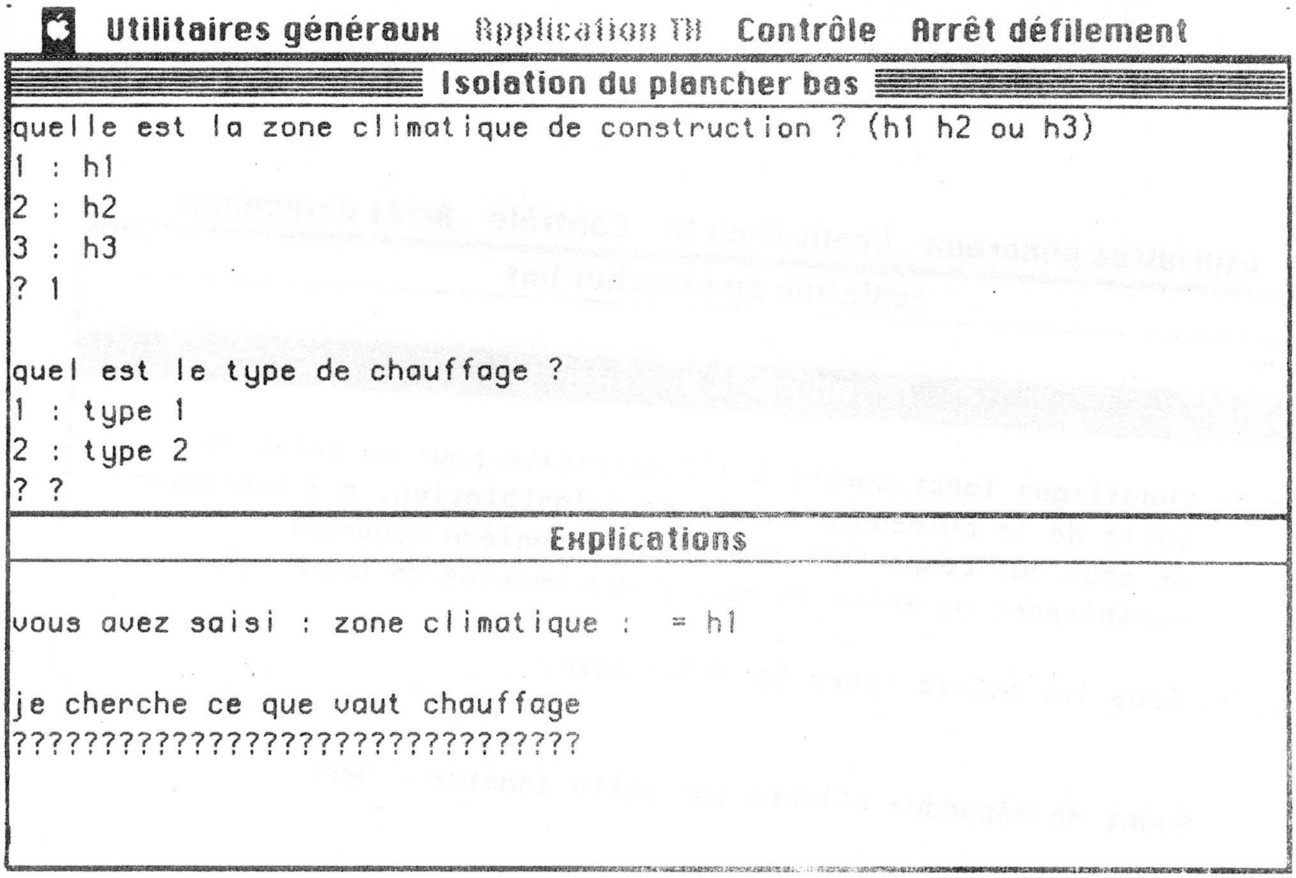


Figure 4. On the same screen : reasoning and explanations.

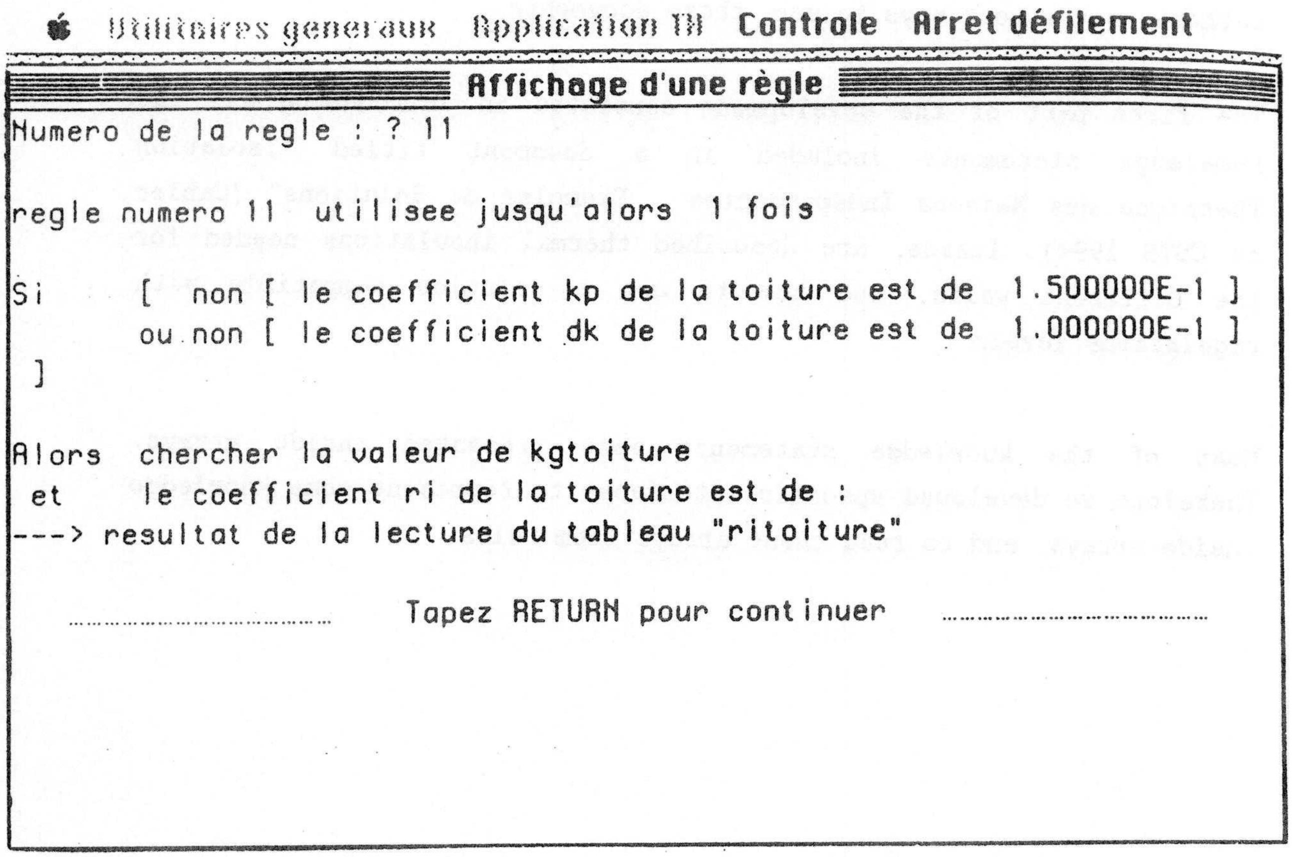


Figure 3. Rule example manipulating coefficients.

The second step of the prototype development is still going on. It concerns the insulating materials, each of them is described in a certificate that gives the material name, its thermal conductivity, its standardized thicknesses, and some insulating characteristics. Certificates will be entered in a data base. We have created a very small similar data base, to test how to make a request in it from a rule.

Concerning this subject, most of the information needed is written. The non written expert knowledge, coming from studied documents authors, gives best ways to use those documents.

The first part of the development consisted in introducing all the knowledge statements included in a document titled "Isolation Thermique des Maisons Indépendantes : Exemples de Solutions" (Cahier du CSTB 1994). Inside, are described thermal insulations needed for the different walls, and how to get a solution compatible with regulations levels.

Most of the knowledge statements were presented inside arrays. Therefore we developed specific interfaces to represent some knowledge inside arrays, and to read these arrays from rules.

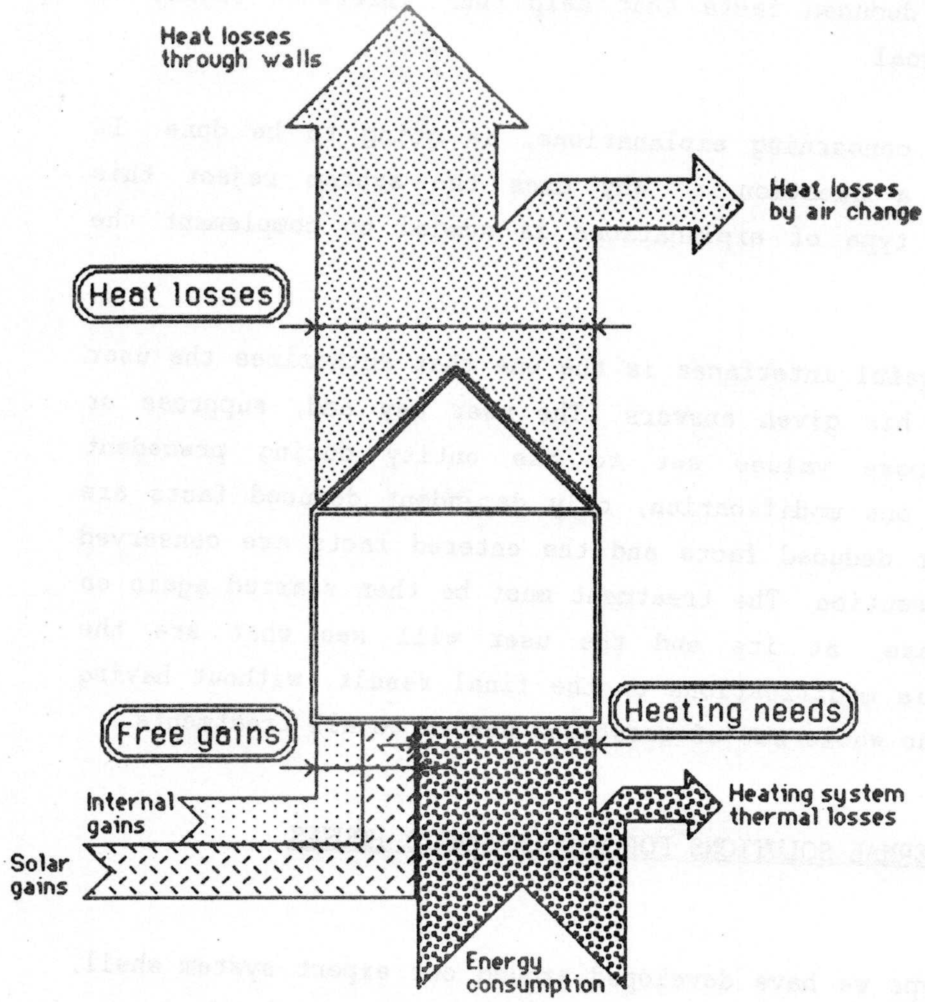


Figure 2 : Coefficient G : global heat loss
Coefficient B : global heating need

Detailed explanations stocks every steps of the last treatment done. This level of explanation is mostly used to debug the software part and the rules base. Non detailed explanations gives an historic of entered facts and deduced facts that help the inference engine to prove its initial goal.

A third interface concerning explanations, is going to be done. It will answer such a question : "why does the system reject this solution ?". This type of explanations is needed to complement the existing types

One of the most useful interfaces is the one that authorizes the user to modify one of his given answers. The user may add, suppress or replace one or more values set to one entity during precedent reasonings. After one modification, only dependent deduced facts are cleared, the other deduced facts and the entered facts are conserved to serve a new execution. The treatment must be then started again on the new facts base, at its end the user will see what are the consequences of his modifications on the final result, without having to answer again the whole set of questions needed to all treatments.

4. EXAMPLES OF THERMAL SOLUTIONS FOR SINGLE FAMILY HOUSES

The first prototype we have developed around our expert system shell, aims at defining thermal solution for a described single family house. A thermal solution is constituted of the global heat loss coefficient, called coefficient G (fig. 2), the global heating need coefficient (coefficient B = G - useful free gains), a set of thermal resistance per each internal and external walls and floors, and the list of insulating materials with their thicknesses corresponding to each resistance. Thermal solutions have to reconcile regulations demands and insulating installation difficulties with the house particularities.

facts base contains all the deduced facts and all the entered facts.

A fact looks like this :

```
        entity1 ( (operator1 value1)
                  (operator2 value2) ... ))
```

The operator allowed are : =, <, >, <=, >=, <>.
These operators works on numeric values as well on strings when they can be sorted.
At least = and <> may compare any types of values.

```
So we may express : building height >= 3
                                    <  5
```

3.3. <u>Interfaces</u>

Menus help the user to command the system. One menu offers reasoning commands that depend of the loaded application. Three other menus present common commands used to manipulate the different bases.

These interfaces do loading, saving, printing, reseting, and displaying actions on files, data, bases, statements... The system offers all these functions for the facts, the rules, the explanations and the statistics.

Statistics are rule counters. There is one counter per rule, it is incremented each time the rule is activated, session after session. Thus, if a counter stays at zero after a great number of various executions, we may ask ourselves about the interest or the good writing of the rule which was always ignored by the reasoning.

Two levels of explanation are available. Explanations are reasoning traces translated into quite a natural language (French actually).

3.2.1. The rules base

The rules base of our expert system shell is made of production rules. A rule is a list of Lisp functions. In the condition part of a rule, two functions are mainly called : the two backward chaining functions. In the action part the usual function used, writes in the facts base a new fact. To be addressed, rules are numbered.

They are represented as follows :

```
(   ( condition1
      condition2
      ...          )

    ( action1
      action2
      ...          )

n)
```

where conditions and actions are Lisp forms directly interpretable, separated by the logical operator : and .

3.2.2. The objects base

What are called objects in this system, do not refer to objects as they are defined in objects oriented language. Our objects are simple descriptions of every entities used inside rules. These objects are used by interfaces as the one which asks the question about one entity. They are also used for the needs of the reasoning, for example a boolean indicating if one entity may get more than one value at a time.

3.2.3. The facts base

It represents the image of the given situation the system has to solve at one moment. It may be filled up before or during the reasoning. The external cladding", and then it examines rules R1 and R2. It tries to answer the question "what are the possible external cladding ?" Later, as the engine wants to prove the first condition of R1 : "the cladding must be an insulating material", it is that complete fact which constitutes the new backward chaining goal. The question to answer is now : "must the cladding be an insulating material ?", and no more a partial question as : "what must be the cladding ?". With this precise goal the engine will try only the rule R3 and not rules R3 and R4.

When one condition can't be verified by activating another rule, the system asks the user about it, if no other fact denies or confirms already the condition.

The backward chaining inference engine working on partial goals is not only used to start a reasoning. For example, in the rules base on thermal solutions, it is used to obtain during the treatment, coefficients values presented as variables of equations.

3.2. The knowledge representation

The knowledge base is constituted of one rules base, one "objects" base, one facts base and a set of functions due to the application. These functions are mostly commands which offer to the user to treat the different problems of the subject.

The knowledge base may also contain descriptions of entities to be sorted, with sort criteria (e.g. external cladding) ; or knowledge statements presented into arrays (e.g. thermal solutions). These arrays can be read by rules, they contain all the knowledge that suit better to such a data representation.

possibly activated the rules which have at least one action that proves the expressed goal.

Example : taken from the rules base on external cladding.

These are 4 rules : R1, R2, R3 and R4 :

R1 : IF the cladding must be an insulating material
AND the wall type is less than III

THEN one possible external cladding is "enduit hydraulique sur isolant"

R2 : IF [or the wall is made of cast concrete
the wall is made of bricks]
AND the wall type is less than IV
AND the fire reaction class is combustible but not inflammable

THEN one possible external cladding is "enduit traditionnel 3 couches"

R3 : IF the thermal insulation is not internal

THEN the cladding must be an insulating material

R4 : IF the thermal insulation is internal

THEN the cladding must not be an insulating material

At the beginning of the selection of possible external cladding, the backward chaining inference engine receive the partial aim : "possible

3. THE EXPERT SYSTEM SHELL : ELSE

The CSTB expert system shell has been elaborated in the "GENIE COGNITIF" department. All programs were written in Le-Lisp, French LISP dialect developed by INRIA (Institut National de Recherche en Informatique et Automatique). It was done on a VAX 11/750, especially dedicated to artificial intelligence applications at CSTB. The software and the two main applications are now implemented on a Macintosh Plus.

This software has been designed to satisfy the needs of our applications, and in the same time try to be the more general it can. In fact the inference engine is able to "think" on different technical problems which aim to select solutions in a set of previous possibilities.

3.1. The inference engine

It executes a backward chaining reasoning. That means that the possible reasoning is controlled by objectives. Two types of treatment are possible : reasoning from one aim perfectly described or reasoning from an aim partly known.

When you start the system, you command the inference engine to make a selection or a choice, which means to find every possible values a given entity may get in a precise situation. So far, the backward chaining inference aim is summarized to a simple entity, and the engine will try to activate every rule which allow the entity to get any value.

During the treatment, intermediate goals are created. They are condition statements of the rules the engine is looking at, trying to activate one of them. These goals are more precise, they are not made of a single entity, but of a complete description of a fact. Indeed, to prove that one condition is settled, it must examine each different component of the condition : entity, operator and list of values. The backward chaining working from a complete goal, only considers to be

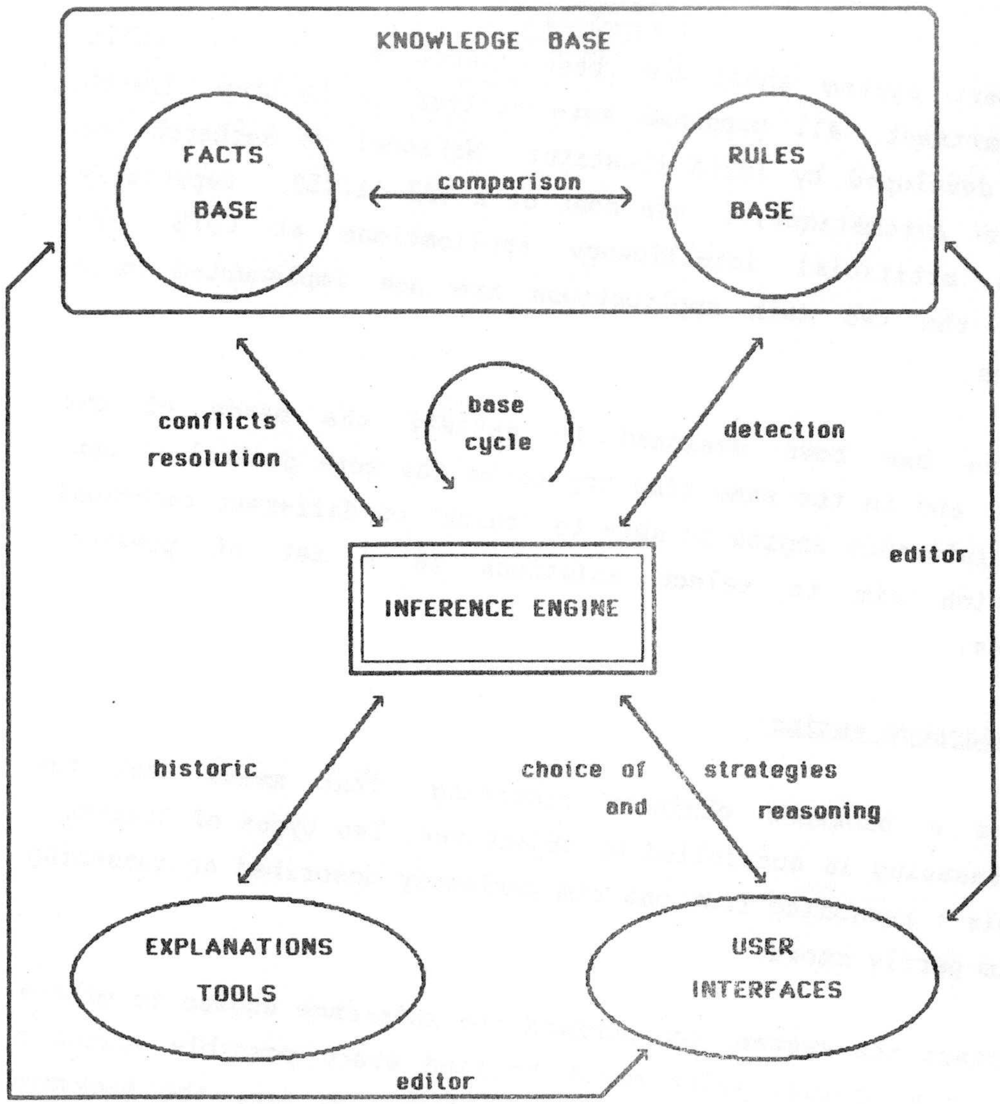


Figure 1. Diagrammatic presentation of an expert system.

Forward chaining mechanism is mostly used to solve designing problems. The backward chaining approach better suits to diagnosis problems in which we try to bear out at posteriori a chosen solution among many possibilities.

2.4. <u>Other elements of an expert system</u>

To be complete and to aim its objectives, an expert system must embed the knowledge base and the inference engine in a set of software tools to make it user friendly. These tools have to facilitate the use of the system and to make understandable a reasoning done.

An explanation tool must exist. It allows during execution to direct the resolution, and after execution to indicate how the system has found the final solution.

To consult and expert system, there must be a high command language using menus, drawings, windows...

A big set of interfaces makes the knowledge accessible to non speciallists, translating it in a as good as possible natural language. It also permits to modify the knowledge base and then controls it consistency.

Expert systems differ from one to another in sophistication of conditions and actions expression : use of variables, quantizers and/or logical operators.

Other forms of knowledge representation exist.Recent research developments of objects oriented language give a range of possibilities to structure knowledge by the description of objects. These objects are the different elements of the application field, their descriptions carry properties, links with their families, heritage mechanism...

2.3. Inference engine

The inference engine is the conductor of an expert system. It is made of logical instructions suitable to knowledge representation and to involved problems, but these instructions do not depend on knowledge values.

The inference engine does cycles. One cycle consist of two phases : the evaluation phase and the execution phase.

Diagrammatically, we may say that during the evaluation phase, the inference engine detects every applicable rules by evaluation of the current state of the facts base. Then during the execution phase, it actions one or every detected rules. The cycles are controlled by a stop-condition : one final solution has been found for example.

Two main kind of logical treatments exist : forward chaining and backward chaining. A forward chaining engine applies rules that their condition part is checked by comparing it with the content of the facts base. A backward chaining engine works the opposite way, looking at the action part to determine which rules may be applicable, and then on these rules it applies the ones which have all their conditions checked.

This characteristic of expert systems, knowledge separate from logic, mostly contributes to their present success, because updating the knowledge base does not touch to logical treatment.

2.2. The knowledge base

The knowledge base is usually composed of two distinct sets :

- The facts base, which contains data specific to the problem treated at a time. By nature this base will evolve with the problem when the system tries to solve it. When the expert system starts, the facts base may be empty or contain a part or the complete description of the situation to be solved ; then the base is filled up by deductions done by the system and by complementary information given by the user as the system asks for it.

- The rules base saves general pieces of knowledge given by at least one human expert when the expert system is built. One piece of knowledge is generally called a rule. Each rule indicates one possible step in the reasoning the expert system may use.

A rule, called a production rule is presented as this :

```
rule nb i

IF   condition 1
 AND condition 2
 AND ...

THEN action 1
 AND action 2
 AND ...
```

"conditions" deal information we hope to find in the facts base, and "actions" usually attempt to modify the facts base.

This software, which is far from commercial computer programs, has already been applied on three substantial subjects :

- examples of thermal solutions for single family houses ;
- aided choice of external cladding ;
- technical control in construction design.

2. EXPERT SYSTEMS : GENERAL PRESENTATION

2.1. Definition and main characteristics

Expert systems are softwares able to simulate human way of reasoning on a given expert field. Their aim is to assist someone to make the best deductions operating on a large bulk of knowledge information.

These softwares use computers storage capacity, and their treatment power, and so expert systems are at least more available and more reliable than best human experts.

As a human expert, an expert system is not only able to end up a complete reasoning on a problem ; it also explains afterwards the way it "thought". And these explanations are clear enough to be understood by a user who is absolutely not an expert on the domain treated.

Expert systems are designed to be used by a non computer scientist as well as a non specialist of the domain.

Their main originality, compared with traditional softwares used to resolve algorithms problems, is to separate distinctively the expert knowledge on the domain treated from logical mechanisms that operate on it. The expert knowledge is set together in something called a knowledge base. Logical mechanisms are written in the inference engine.

very well to algorithmical treatments. So far there are not a lot of classical software tools offered today to help building professionals to use this type of information. That is where and why expert systems may be useful to the building field.

CSTB has been developing, for two years now, some expert system prototypes applied to various sectors of the building domain. Our study on expert systems in construction aims to appreciate and to demonstrate what might give this new computer science technique in resolving building problems.

This survey leads CSTB to elaborate one expert system shell on which we applied different building fields. The main characteristic of expert systems is the complete distinction between the software part and knowledge information it manipulates. It is the knowledge representation that links software tools to the expert application. And so, one same reasoning software qualified as "intelligent" can "think" on different subjects. Just one restriction to the diversity of problems one software can treat, is due to the knowledge representation chosen in which every subjects have to be modeled. The more the knowledge representation is varied and sophisticated, the more the software to work on it is long to be developed, the more the range of possible application themes is large and the more the prototype is long to perfect.

For our first needs in expert systems prototyping, we have written a little software that works on a quite simple representation. It allows to develop easily with rapidity little prototypes. They are just big enough to determine the interest and the difficulties of one subject. So it gives soon a realistic idea to experts and potential users of what an expert system can do on their own problem.