

# **Aided Diagnose of Structural Pathologies with an Expert System**

Ernest Bernat<sup>1</sup>, Lluís Gil

*Polytechnic University of Catalonia – BarcelonaTECH. Department of Structures and Material Resistance. ETSEIAT – Terrassa (Spain)*

## **Abstract**

Sustainability and safety are social demands for long-life buildings. Suitable inspection and maintenance tasks on structural elements are needed for keeping buildings safely in service. Any malfunction that causes structural damage could be called pathology by analogy between structural engineering and medicine. Even the easiest evaluation tasks require expensive training periods that may be shortened with a suitable tool. This work presents an expert system (called Doctor House or DH) for diagnosing pathologies of structural elements in buildings. DH differs from other expert systems when it deals with uncertainty in a far easier but still useful way and it is capable of aiding during the initial survey ‘in situ’, when damage should be detected at a glance. DH is a powerful tool that represents complex knowledge gathered from bibliography and experts. Knowledge codification and uncertainty treatment are the main achievements presented. Finally, DH was tested and validated during real surveys.

Keywords: Expert System; Buildings; Maintenance; Inspection; Diagnosis; Assessment.

## **1. Introduction**

During the past decade construction has been a driving force of the economies of many countries. This surge has created an increasing volume of buildings that will have to be

<sup>1</sup>Corresponding author: [ernest.bernat@upc.edu](mailto:ernest.bernat@upc.edu). Colom 11 TR45 D.137, Terrassa 08222 Spain.

maintained in the near future. Moreover, nowadays the industry is slowly changing from classical construction towards sustainable building. Therefore, maintenance and renovation are processes that enlarge the life-cycle of buildings. In short, most of the recently constructed building will have to be surveyed and/or repaired during the following years. Taken together, these factors point to an explosion in the demand for qualified personnel for diagnosing pathologies of buildings, infrastructure and other forms of construction—of which there will be a greater number in use and to be conserved.

This backdrop is the motivation for the development of a program to provide support to the new inspectors who will have to diagnose the pathologies of this immense volume of buildings.

This type of program is not intended to substitute technicians; rather, its main objective is to assist in the training of new experts and provide them with greater confidence in the most complex diagnoses. Furthermore, it helps free these highly qualified workers from mundane tasks, enabling them to focus on the most complex cases.

Described here are the steps followed in the development of a first operating prototype of the software. It was focused on buildings for several reasons, including their great economic weight, user demands and the large volume of buildings already constructed. Another reason was to obtain a more useful tool: buildings generally have a lower inspection level than large public works, therefore they were considered to suffer more damage.

Knowledge codification, which includes obtaining the information, ordering it and codifying it in form of objects and logical rules, was the most challenging part of the investigation together with the necessary uncertainty treatment. How these tasks have been carried out is fully detailed in their corresponding sections, but in general lines the first one was overcome with the support of a team of experts from private companies and university whose experience, which covered all structural elements of a building, was classified. The second one was faced by simplifying the uncertainty levels for not losing the qualitative approach and still

offering a reliable and transparent tool. Classifying symptoms (observable information) into required ones and recommended ones was an original improvement.

This paper is divided as follows. First, past and present artificial intelligence (AI) programs developed for the construction field are described. The objectives of developing DH are then outlined. Next, expert systems are briefly described and are evaluated in the context of other methods. The process employed for developing DH, and the results of initial trials, are then explained in detail. A brief qualitative comparison with similar tools is also included and, lastly, conclusions are presented.

## **2. Review on Artificial Intelligence in construction and Aim of the work**

The term artificial intelligence (AI) was first coined by John McCarthy at a conference in Dartmouth in 1956. Since then, AI has been used for a myriad of applications, including modelling and decision making in the construction field.

One of the most important milestones in the practical use of *intelligent* reasoning was the development of expert systems, which are explained in more detail below. Briefly, an expert system can be defined as a computer program that is designed to emulate a human expert for solving problems in a specific area of knowledge as it is explained in Harmon and King (1985) or Giarratano and Riley (1994).

The recent advent of artificial neural networks, particle swarm optimization method or genetic algorithms have provided a new practical stimulus for AI in the field of decision-making around engineering activities. Nowadays, AI includes a lot of different fields, like robotics, representation of natural language, human reasoning and others, but the most commonly used in construction are artificial neural networks, Case-Based Reasoning (CBR) and expert systems. However, recent investigations on genetic algorithms like Cheng et al. (2002) and particle swarm method presented in Zhang and Xing (2010) or Jiang et al. (2011) among many others, suggest them to be widely used in the development of the future most complex decision-making tools.

Although construction was one of the last fields to use this technology, see Kaetzel and Clifton (1995), over time several tools and lines of research have since been developed. These include:

- Support for decision making in project design and planning: *e.g.* CONBPS (Construction Best Practice System) for modelling of the construction process by Poon (2004); neural networks to calculate the shear bearing capacity of FRP – fibre reinforced polymer – strengthened reinforced concrete beams by Tanarslan et al. (2012); or for the assessment of construction environmental impact in Zhao et al. (2006).
- Support for decision making in the building process: *e.g.* OLSC (On-Line System for Construction) for multi-criteria analysis presented by Kaklauskas (2007); CoSPES (Construction Site Preparation Expert System) for construction tasks written by Albert and Wu (1997); CONBPS by Poon (2004); for construction project supplies there are many example like Kumaraswamy and Dissanayaka (2001); for determining the rippability of rocks in Tiryaki (2007); for adapting tunnel construction in base of its displacements using Support Vector Machine (SVM) together with a particle swarm as shown in Jiang et al. (2011); or a fuzzy multi-objective particle swarm to optimise time-cost-quality in construction presented in Zhang and Xing (2010).
- Support for the diagnosis and repair of construction pathologies: *e.g.* MDDS (Masonry Damage Diagnostic System) for diagnosing pollution caused pathologies in masonry projects proposed by Van Balen (1996 and 2001); RCDES (Reinforced Concrete Diagnosis Expert System) for diagnosing reinforced concrete structures in Peter (1996); REPCON (Repair of Concrete) for repairing concrete structures written by Moodi and Knapton (2003); concrete bridge management by Brito et al. (1997); DIASYN (Diagnosis Synthesis) for concrete bridges used in Zhao and Chen (2002); maintenance of cement bridge decks as in Yehia et al. (2008); for evaluating bridge damage in Lee et al. (1999); for diagnosing cracks in reinforced concrete in Chao and

Cheng (1998) and Yeong et al. (2007); and for diagnosing corrosion of pipes included in Najjaran et al. (2006);

These have been applied to buildings: *e.g.* for evaluating damage to buildings from cyclones in Murlidharan et al. (1997); for diagnosing framed reinforced concrete structures in Lu and Simmonds (1997); for diagnosing building defects in Koppány (2005); SCAMS (Subsidence Case Management System) for evaluation of pathologies due to soil settlement presented in Anumba and Scout (2001); and for diagnosing seismic damage in metallic structures like in González and Zapico (2007).

Many of the expert systems for construction that were developed up to the early 1990's can be found in Kaetzel and Clifton (1995). A list of expert systems for construction project planning can be found in Yamazaki et al. (1990).

Obviously, when the problem to be solved is very specific it can be handled in greater depth and the results will be more accurate. However, it was observed that there was no software developed for the global inspection of buildings, as all available programs focus on a specific material or structural type. So, having observed that there was no generic building-inspection software on the market, we sought to develop an operative prototype in order to prove the technical feasibility of such program. Our initial objectives comprised: a) Researching, summarising and normalising the symptoms associated with each building pathology; b) Developing a prototype of a knowledge-based expert system for diagnosing building pathologies, including ordering, encoding and programming of symptoms data according to the programming tool chosen; and c) Evaluating the resulting software, and then generating conclusions.

### **3. Used tool: Expert Systems**

Expert Systems (ES), which fall under AI, are computer programs that are designed to imitate human experts in solving problems from a very specific area. Modern expert systems bear little resemblance to their earliest predecessors, which arose in the 1960's and which were originally

designed to solve general problems (Harmon and King 1985). They have three fundamental components: an inference motor, a knowledge base and an interface.

The inference motor is in charge of obtaining new data based on the existing data in the knowledge base and that provided by the user. It can control this data collection process according to various criteria, including forward reasoning and backward reasoning. The forward reasoning is based on performing a basic search to determine which data is needed for continuing with the reasoning. Albeit this leads to a disordered and chaotic appearance from the user's perspective, it allows execution of planning tasks for which solutions are not known *a priori*. In contrast, the backward reasoning performs a deep search which is focused on an objective (a hypothesis which is to be verified) and determines the information that must be confirmed in order to validate the chosen line of reasoning. This method of inference is much more natural, logical and ordered from the user's perspective, and is especially suited for cases in which there are a limited number of known solutions, as it is diagnosis of building pathologies.

The knowledge base stores the information from the human expert that the program may use in searching for a solution. The knowledge base is encoded differently in each expert system, but the most common and powerful way is through objects, which are related to each other by production rules (*i.e.* through *if/then* logic rules, which are associated with the inference via *modus ponens*, see below).

Finally, the interface is a crucial part of an expert system, as it must be adapted to allow easy entry and modification of information from the knowledge base and enable justification of the diagnosis. Moreover, it has to be user-friendly, since it is used for data entry by the user and for the delivery of results.

For greater detail on the structure and function of expert systems, the reader is referred to Harmon and King (1985), Giarratano and Riley (1994) and Kaetzel and Clifton (1995).

### 3.1. *Why an expert system?*

The decision to employ software in the form of an Expert System was based on the ease of programming of this method as compared to conventional programming. This selection was also done because ES show a more natural and predefined solving behaviour than artificial neural networks (ANN) whose drawbacks are shown below. Moreover, ES are far simpler and more organised than genetic algorithms (GA) or particle swarm optimisation methods (PSO) which could work together with support vector machines (SVM) as it was done in Jiang et al. (2011).

The main advantages of an expert system over conventional programming comprise:

- The possibility of completely separating the solution method for the problem from the data, which allows ready modification, correction or expansion of the knowledge base, as well as creation of shell-type tools, which already contain the inference motor and only require the knowledge base;
- The use of symbolic programming, which allows incorporation of symbolic variables (*i.e.* words from natural language, such as “*crack*”, “*oxidation*”, etc.), between which logic-type relationships can be established (*e.g.* *if/then* production rules, or qualifiers such as *and*, *or*, *equal to*, *different than*, etc.).

Just as other programs based on knowledge and experience, the work described here adjusts better to the programming patterns of expert systems than to those associated with classic programming, which is focused on the use of numbers and the procedural solution of problems.

The main advantages of an expert system over ANN in diagnosis of building pathology comprise:

- Expert systems receive information from a human expert (once that information has been properly ordered and encoded), whereas neural networks learn by examples. Therefore, for a field such as building pathology, in which there are so many possible combinations and symptoms, an extremely large number of cases would be required in order to train a neural network.

- Neural networks automatically establish the relationships between variables from the examples provided. This means that solutions that are initially unknown to the expert can be found. In the field of diagnosis, this possibility is counter-productive, as the aim is to provide confidence in the identification of the cause of damage, rather than to “discover” pathologies based on atypical symptoms. Furthermore, the automatic encoding of knowledge complicates its subsequent modification since the information introduced does not remain structured, but rather is stored diffusely throughout the network. In fact, several authors have referred to neural networks as a “black box” whereby the relation between each input and its corresponding output remains obscure to the user as González and Zapico (2007) pointed out.

For these reasons, neural networks can be considered more amenable to problems that are poorly structured or that feature non-explicit knowledge—the very opposite case to building damage diagnosis, in which the causes and effects are well known and can be easily correlated.

Other actual tools, like SVM could overcome the extra-learning problem of ANN. They base their learning on statistics in order to solve multidimensional functions whose domain is previously delimited as explained in Jiang et al. (2011). However, its working procedure is “hidden” from user in the same way it was in ANN, so it is not suitable in developing a tool aimed to aid new professionals and to help them in their formation.

Like ANN, GA and PSO are mainly oriented to solve problems from incomplete and bad-structured information. Cheng et al. (2002) used GA to deal with the problem of flood prevision which is a clear example of incomplete and bad-structured data. Modelling algal blooms in Muttill et al. (2006) or trying to foresee sunspots as shown in Xie et al. (2006) could be another two examples. As structural damage diagnosis at first visual inspection has a relatively little field of possible solutions and related data can be organised, GA and PSO are not as suitable as ES are to deal with this problem.



### **3.2. *Expert Systems drawbacks***

Today, Expert Systems are not widely used. Lately, this programming tool has been progressively substituted by CBR (Case-Based Reasoning) firstly and ANN (Artificial Neural Network) recently, due to its difficulty to structure and validate the general expertise of most of the nowadays common problems. Nevertheless, ES are still useful in some fields, like diagnosis of building pathologies (or detection of the damage's cause), in which knowledge is very well organised, and the development of an expert system is easier than in other fields.

The use of an expert system can be seen as step back, but in our opinion, it can break the current trend, which is turning to numerical treatment of any problem. It loses touch with the real problem, and all is reduced to adjusting many factors in order to reach the expected solution whereas the logical reasoning process and the justification of the obtained solution are absolutely forgotten. As there is the aim to develop a tool that could be also useful at training new human inspectors for diagnosing structural damage, it seems suitable for this tool to be transparent, to use qualitative information rather than quantitative coefficients decided by a group of experts and the most important, to have the possibility of giving an explanation about how the conclusion was reached in basis of the information and the logical inferences.

## **4. Developing the prototype. Methodology**

Outlined below is the process followed to develop the prototype of the new tool for construction damage diagnosis, from the scope delimitation to the translation possibilities, including the knowledge codification and the uncertainty handling.

### **4.1. *Focusing on a specific area***

Once the problem was centred on building pathology, it was decided that the level of casuistry should be minimised, since the objective was merely to create a prototype to prove the technical feasibility of this type of program. It was ultimately decided that the problem would focus on pathologies that affect structural elements or that directly derive from these elements. This

choice was due to the fact that the principal repair costs in building pathology are those related to structural damage as pointed out in Calavera (2005) or by The Masonry Society Construction Practices Committee (2004).

Different pathologies were not treated with the same level of depth; the south-European most common type of structure was emphasised: a framed concrete building with dividing masonry walls. Special attention was paid to the principal structural elements: pillars, beams and floor slabs.

Altercations to masonry elements were also given special attention, as the number of these pathologies is much higher than other types, and they are generally caused by the malfunction of the structure of a building. In terms of pathology types, cracks are the most common form of damage among structures, as presented in Rodriguez (2000). As such, they were studied in more detail than other symptoms.

Because of the aim of DH, the range of pathologies to be diagnosed by the expert system was limited by including only data that could be obtained through visual inspection of a building with the naked eye; complex assays were therefore omitted. Thus, the resulting software was honed for providing a preliminary qualitative inspection in a transparent way.

#### **4.2. *Choosing the programming tool***

A shell-type program—which already contains the inference motor and the interface—was chosen as the development platform for the prototype. This choice was based on various factors, including the knowledge (data) itself to be introduced as well as the scope of the work at hand (for a first prototype).

The information collected on symptoms was generally well ordered. For this reason, it was preferable to use encoding based on objects and values. This way, the tool had to use inference through *modus ponens* and relate objects with production rules. *Modus ponens* is the most widely used inference method. It establishes if/then type of relationships. Hence, once a

rule has been introduced ( $A \rightarrow B$ ), and the left-hand side factor (A) has been confirmed, the result provided by the inference is affirmation of the right-hand side factor (B).

An object is the symbolic representation of a characteristic, fact or real element (*e.g.* “direction of the crack”). Each object can adopt different values (*e.g.* [*vertical, horizontal* or *diagonal*] for “direction of the crack”).

Considering that the developed expert system is a diagnostics program aimed to train novel inspectors in on-field preliminary inspection, the inference process had to be ordered according to user needs, and consequently, the shell had to allow backward reasoning. The shell also had to be monotonic: the inference process could not be allowed to change the information introduced by the user, since the program would then be diagnosing a different damage from that observed. Furthermore, given that the developed software was designed to simplify the diagnostic process as well as the data required, the use of either numerical uncertainty or diffuse logic was ruled out as it is outlined at subsection 4.4.

Commercial software Acquire 2.1 was selected from among the various software packages that were available at the time of development, as it responded to all of the programming needs for the work at hand and has features which are highly amenable to the creation of a prototype of a small expert system.

### **4.3.      *Encoding and introducing the information using Acquire 2.1***

The first step in encoding the information comprised classifying the pathologies according to the structural element affected and the constituent material of that element. Many small and focused expert systems (modules) were created in this way, each one for dealing with the diagnosis of pathologies of one structural element or one material. Most of the information related to the symptoms of the building pathologies was extracted from documentation like Calavera (2005) and The Masonry Society Construction Practices Committee (2004), and from some experts interviews.

In total, seven specialists were interviewed. Two engineers from the Construction Engineering Department, two more from the Structures and Material Resistance Department, all from the Polytechnic University of Catalonia – BarcelonaTech, plus two architects and another engineer from the company Asistencia Técnica Industrial S.A.E, ATISAE, who are expert inspectors. The aim of the interviews and meetings was to get a deeper description of selected pathologies. They were also oriented to improve the knowledge acquired from bibliography and to obtain a more confident response. As all pathologies included in DH are common and can be diagnosed at a glance by a professional, the team of experts did not show any discrepancy about the diagnosis nor the causes of damages. This is one of the reasons, as it is widely shown later, why fuzzy logic and numeric evaluation of uncertainty were not considered in the development of this software whose purpose is to help inexperienced inspectors at diagnosing common structural pathologies of buildings.

The way the knowledge was acquired is following described. First of all, the major part of it was extracted from bibliography. After this, an expert from the Structures and Material Resistance Department was selected as the conductor of the interviews and the meetings. The first meeting was set to discuss the scope of the selected damages and the main variables that had to be included, distinguishing between the defining symptoms and the ones that do not need to be observed to come up with the diagnosis. Then, a preliminary version of the prototype was implemented and individual interviews were carried out with each expert in order to improve the result. As no contradictions between experts were observed, every opinion was taken into account. The information given by the experts in the individual interviews was complementary and never confronts other's advice. After improving DH, a full meeting was arranged to present the result to the experts and discuss about the last possible improvements that mostly affect superficial things like the interface language or the distribution packaging. The knowledge base was consolidated after this meeting.

Because of the scope and the aim of DH it was not considered to deal with uncertainty in an analytical way. Moreover, no discrepancies between experts' recommendations were observed,

so quantifying uncertainty in common pathologies would not be worth the effort as the diagnosis were clear. For these two reasons it was decided that the selected team of experts was enough to provide the information needed. For further research including more complex pathologies, uncertainty would have to be taken into account in a numeric way because different opinions are expected in less common problems, so a bigger group of experts (statistically significant) would be required in order to value this uncertainty in the conventional quantitative approach.

Once the pathologies had been classified, the next step was to encode the symptoms of each one by transforming the collected information into objects and rules, which are schematically represented in Figure 1.

The objects created for encoding information in DH can be classified according to its function. First of all, there are starting objects (continuous striped rectangles in Figure 1) which are symbolic representations of the symptoms. When the expert system asks the user about a particular symptom, its answer will be reflected in the value taken by the corresponding starting object. Secondly, there are auxiliary objects (discontinuous horizontal striped rectangles in Figure 1) whose main function is to give internal structure to the program, thus providing a more logical and ordered presentation of the inference for the user. The use of these objects makes heredity possible, so at the moment at which the inference process assigns a value to an auxiliary object, this value implicitly represents all of the information provided up to that moment by the user over the course of a DH query. Finally, there are conclusion objects (spotted rectangle in Figure 1) which symbolically represent pathologies that can be diagnosed by the expert system. At the moment these objects adopt a favourable value (either “Yes” or “Yes, but”, as explained below) a diagnostic is reached.

In terms of the relationships among objects, Acquire 2.1 can use production rules (Figure 2) and action tables. These two methods only allow a single value to be assigned to a unique object as a conclusion of the relationship. Production rules are best suited for cases in which many objects are considered. These objects should adopt very specific values in order to reach a conclusion. This is the case of the last step indicated by the dashed lines in Figure 1,

where it can be observed that, for a given combination of values for some different objects, one value is obtained for the conclusion object. In contrast, action tables are indicated for cases in which very few objects are considered and in which nearly all of the combinations of values of these objects are valid for generating a conclusion in the inference. Roughly speaking, an action table can be considered as a set of production rules grouped together as a matrix. Action tables are used in practically all of the relationships that assign a value to an auxiliary object. Each step in Figure 1 that is marked with a solid line corresponds to an action table. A complete diagnostic tree, following the encoding process described, is shown in Figure 3.

#### **4.4.        *How to handle uncertainty***

DH considers uncertainty, which is intrinsic to any type of heuristics-based reasoning. In general, uncertainty might be encountered in the inspection as well as in the logic rules employed for diagnosing the pathology. However, it has been shown previously that there was no uncertainty in the logic rules for DH as experts and bibliography did not show discrepancies in their diagnosis. It is because DH is aimed to help at diagnosing well-known cases to train new inspectors or free highly qualified workers from ordinary activities. Despite the fact that most of the current expert systems handle uncertainty through inexact reasoning and fuzzy logic –examples of this are Zhao and Chen (2002), Murlidharan (1997), Lu and Simmonds (1997), Chao and Cheng (1998), Yeong et al. (2007), Najjaran et al. (2006) or Anumba and Scout (2001) – the prototype presented here was designed to treat uncertainty qualitatively, which is simpler and more transparently for the user than the conventional quantitative approach. It allows DH to be used as a learning tool. Actually, this approach goes back to the empirical treatment used in the original expert systems. The expert system was thus conferred with a tool that enabled it to deal with uncertain information, although with no complex mathematics required, as these tend to overcomplicate the justification of the obtained solutions to the user and make them less transparent—when these explanations are in fact one of the main advantages of expert systems. This is because current numerical treatment is based on the use of

coefficients that are chosen by experts according to their personal experience (in order to represent the scenario that certain symptoms correspond to a specific pathology) and these coefficients are usually hidden to users, who do not receive all the inference information. Instead, showing all the mathematically represented uncertainty information to the user is not the best solution either as the amount of data would make it difficult to understand the reasoning procedure anyway. This, together with the nature of the problem faced in this research lead us to develop a prototype with a different system that would be more general, simpler and more intuitive. As a drawback, it would not give any numerical value to describe the confidence on the solution. Therefore, the empirical treatment of uncertainty used here is not based on a scientific approach like probability. We assume the limitation of this approach and admit this technique could not be suitably extended to the development on an expert system that had to deal with more complex cases in which conflicting views between experts were likely to rise. In this other situation, fuzzy-logic would be likely to face the problems with more success than the original method proposed here. In conclusion, it could be said that the way the uncertainty is treated in DH is the best option for the specific faced problem and the particular aim of the program but it could not be the best for developing tools oriented at helping the highly-qualified professionals at taking the most difficult decisions.

The goal for representing uncertainty in DH was to establish two levels of confidence in the solution. This was based on the fundamental hypothesis that there are two levels of symptoms in any pathology: Required ones and Recommended ones. Required symptoms are those which are most characteristic of a particular type of damage and which must be observed in order to diagnose it, whereas Recommended symptoms are those whose observation—though not required for diagnosis—allow to reach a greater confidence in the diagnosis. This treatment of uncertainty was encoded by assigning one of either two values (“Yes”, or “Yes, but”) to the conclusion object which confirms diagnosis of a pathology with different levels of confidence. If all of the observed symptoms correspond with the expected symptoms, then the conclusion object takes the value “Yes”, and consequently, the diagnosis is confirmed with maximum

confidence. If any of the Recommended symptoms are missing, then the conclusion object takes the value “Yes, but”, and therefore, the diagnosis is made with less certainty. In this last case, the results show which of the Recommended symptoms are missing. In contrast, if one of the Required symptoms is not observed, then the pathology is not diagnosed. These three scenarios are summarised in Table 1.

<i>Scenario</i>	<i>Result</i>
Required symptoms = expected values Recommended symptoms = expected values	Diagnosis with maximum confidence (Value “Yes” in pathology object)
Required symptoms = expected values Recommended symptoms $\neq$ expected values	Diagnosis with less confidence (Value “Yes, but” in pathology object)
Required symptoms $\neq$ expected values	Undiagnosed pathology (Value “No” in pathology object)

Table 1. Combinations of symptoms shown with their corresponding levels of diagnosis confidence

It should be mentioned that all of the auxiliary objects are associated with Required symptoms, and that the Recommended symptoms are the last ones that the user is asked about; hence, they are incorporated in the last step of the inference, the production rule. Along the development process, it was stipulated that any symptom which exhibits a high degree of variability in the real world, or which is difficult to identify, can never constitute a Required symptom for the diagnosis. A clear example of this would be the width of cracks, which, although indicative of the magnitude of the damage, is rarely meaningful for determining the cause of the pathology.

#### **4.5. *User interface and translation possibilities***

In the developed prototype, the user interface was limited to two environments: data entry by the user (as requested by the program through a series of window pop-ups) and the presentation of results in the form of a report which includes either the diagnosed pathology or a course of action in the event that no diagnosis was obtained.

Furthermore, the entire expert system’s interface can be easily translated into any language by incorporating additional objects and relationships that establish one-to-one



relationships between the new inputs and the original ones, which consequently become auxiliary objects.

## 5. Running DH: How does it diagnose?

Since the development of DH has been detailed, now here it is presented how it works and which its main features are through the analysis of a real case. However, the basic steps the program always takes should be explained first.

Observing the flowchart in Figure 4, it is noticed that at the beginning of the diagnosis the user have to provide, at least, the information related with the structure typology and material and the kind of affectation. After this, the program chooses a possible diagnostic and tries to confirm its main symptoms (those called “Required” in the section 4). If some of these symptoms are not confirmed by the user, DH changes its assumption and tries to confirm another pathology that actually matches with the information already provided. If there are no pathologies in the knowledge base that agree with the observed main symptoms, DH cannot diagnose the affectation. Else if one possible diagnostic is confirmed by the observation of its main symptoms, the program follows to the next level and tries to be fully certain of the diagnostic that has been reached. At this point, more specific symptoms (previously called “Recommended”) are asked to be observed, and if the inspector-user confirms all them, DH achieves a fully certain diagnostic. If not, the diagnostic is anyway presented but with no fully certainty. In this case, an explanation with which the secondary expected symptoms were is also showed.

For better understanding, a real diagnostic case is outlined below. Each step in decision-making progress is explained, and both the questions of the expert system and the answers from the user are indicated. There are also pictures and sketches of the analysed pathology obtained in the field inspection in cooperation with the company ATISAE.

<i>Step</i>	<i>Question</i>	<i>Answer</i>
1 <sup>st</sup>	Structure typology and material	Concrete Beam (*)
2 <sup>nd</sup>	Kind of pathology	Crack (*)
3 <sup>rd</sup>	When does the problem become visible?	In use (**)

4 <sup>th</sup>	Where are the cracks located in the beam?	End (*)
5 <sup>th</sup>	How is the crack orientated?	Inclined (*)
6 <sup>th</sup>	Is the beam supporting a stairs or a cantilever? Or is the beam placed in the edge of a floor?	No
7 <sup>th</sup>	How many cracks do you see?	A few (2 to 4) (*)
8 <sup>th</sup>	Has the beam been constructed in two times? If you select Yes, means that it's a beam constructed in two times or that it's a continuous beam constituted by two simple beams that have been united in situ.	No (*)
9 <sup>th</sup>	Is the end of the beam connected with a rigid node? If you choose Yes, it is assumed that the analysed element is a span of a longer continuous beam.	No
10 <sup>th</sup>	Where in the section are the cracks located?	Around the upper head (*)
11 <sup>th</sup>	How wide is the crack?	Variable (*)
12 <sup>th</sup>	How deep is the crack?	Cross the beam (°)

(\*) See Figures 5 and 6

(\*\*) It was reported by the building users

(°) Assumption from the fact that the same cracks are observed in both sides of the beam (see Figures 5 and 6)

Table 2. Summary of the running for a case diagnosed by DH

The analysed building is a framed, reinforced concrete structure that comprises four floors plus a basement level. The damage was observed in a secondary girder on the 2nd floor.

Figure 5 and Figure 6 are pictures of the crack to diagnose from either side.

Once defined the problem, now the different steps DH took are summarized in Table 2. First of all, the user selected the structure typology and material (concrete beam) and the kind of pathology (a crack). From this point, DH automatically selected a suitable pathology (any among all possibilities that could explain a concrete beam cracking) and tried to confirm it. At each of the eight following questions, DH could have been changing its initial assumption to a pathology that definitely agreed with the defining symptoms introduced by the user. For example, the question in 6<sup>th</sup> step (Table 2) is oriented to confirm or discard one of the possible causes for inclined cracks at the end of a concrete beam: torsion. In the same way, questions in 8<sup>th</sup> and 9<sup>th</sup> steps tried to find out if the pathology was associated with the construction procedure or with an excessive negative momentum at a rigid node respectively. The next steps (11<sup>th</sup> to 12<sup>th</sup>) and questions DH made were only to add certainty to its decision of a shear failure. So, finally, DH reached a diagnostic and presented a report with the pathology and also added a

warning due to the high danger level of this kind of pathology. A screenshot of the program running is presented in Figure 7.

In conclusion, Doctor House diagnoses the case as shearing cracks due to excessive stress in the concrete. This result is totally consistent with analysis of samples extracted from the building, which revealed that the resistance of the concrete used was much lower than that recommended for this type of structure. The program also warns of sudden collapse, which is associated with said pathology. Lastly, the program's results were further supported by the conclusions of a team of experts in the field of building pathology, who concluded that the cracks were indeed due to excessive shearing force.

## **6. Results, Discussion and Comparison with similar Expert Systems**

Once the prototype of DH was finalised, its functions were evaluated in a series of tests. The initial round of test cases was taken from the literature. Especially, from Muñoz (1988) which summarises many cases on real pathologies diagnosed by human experts. In order to ensure objectivity, a different source of information was used for this evaluation than that used for the programming of the expert system. The results obtained (Table 3) are satisfactory and the correct pathology was diagnosed in most of the test cases. Only a few discrepancies were observed and they were the result of missing information in the test case example, which was limited to a photograph and an expert diagnosis description in the corresponding book. In three of the cases summarised in Table 3 the result was not fully satisfactory because it exceeded the knowledge limit of DH. It is that the pathologies to be diagnosed were out of the scope of the expert system. It is not such a big problem as DH is a prototype which was developed just to prove the technical feasibility of this kind of software. On the other hand, the cases for which the statement “Insufficient information of the test example” is displayed in Table 3 need further justification. For all these cases the problem was that there were validation cases from bibliography so access to the real structure was impossible and the available information was limited to a few pictures and a brief description of the problem that was not enough to provide

the expert system required information to achieve a correct and reliable diagnostic. For example, the torsion of concrete beams was not diagnosed with full confidence because the beam position related to the stairs or cantilever elements was unknown. In the same way there was no information about the cracks in the walls of the lower floors to give plenty confidence on the diagnostic of differential settlement of a pillar. The lack of an accurate description of the cracks at the down side of the slab made it impossible to diagnose the punching cause without any doubt, and finally, there was no information about the construction procedure that might be the basis for the identification of an initial thermal reaction as the problem that caused the 10<sup>th</sup> pathology summarized in Table 3. All these problems arisen from the impossibility to have direct contact with the structure would not be found in real inspection like the one fully explained at the previous section because the access to the building visible part of the structure should be granted.

<i>DH Module</i>	<i>Pathology</i>	<i>Result</i>	<i>Problem</i>
Reinforced concrete pillars	Hydraulic retraction	-	Knowledge limit exceeded
	Excessive tensile stress	+	
Concrete beams	Torsion	~+	Insufficient information of the test example
	Bending	+	
Masonry elements	Differential settlement of a pillar	~+	Insufficient information of the test example
	Absence of joints	+	
Slabs	Punching	~+	Insufficient information of the test example
	Excessive bending of a beam	~-	
Short reinforced concrete cantilevers and nodes	Horizontal tensile	+	
Reinforced concrete windcreens and walls	Initial thermal retraction	~	Insufficient information of the test example and Knowledge limit exceeded
Facades	Map cracking	+	
Metallic elements	Generalised corrosion	~+	Knowledge limit exceeded
Wooden elements	Wood worm	+	

Table 3. Results of the first round of evaluation tests

A second round was employed to validate DH with real cases (together with ATISAE Company). Results from all 5 tests on damages affecting concrete beams and masonry walls

were successful and the diagnosis coincided with the one made by another unassisted inspector. One of these tests has been explained in section 5 as an example.

The final result is a decision-making program that comprises ten modules that can diagnose pathologies associated with reinforced concrete pillars; concrete beams; frameworks; nodes in framed concrete structures; short reinforced concrete cantilevers; masonry elements; reinforced concrete windscreens and walls; facades; foundations; metallic elements; and wooden elements. However, it should be noted that in the initial prototype, the last three modules have a rather limited scope. In total, more than 900 objects and over 1,200 rules were required to develop the software, which can diagnose over 200 pathologies of building structures. The results obtained in the various phases of evaluation are totally satisfactory, indicating that the prototype of DH can diagnose dozens of pathologies with a level of certainty greater than 80% (in agreement with the tests performed). A great part of the errors detected were due to the knowledge limit of this prototype or the lack of in situ information. Furthermore, the level of certainty grows up to 90% if all of the information required for answering the questions is available.

Finally, if the main features of DH are compared with the capacities of other Expert Systems in the field of maintenance of buildings some conclusions arise. First of all, DH is an expert system that deals with the diagnosis of structural pathologies in different structural elements and materials, whereas most of the similar expert systems focus in one specific structural element, eg. concrete bridge decks in Yehia et al. (2008), or one specific pathology, eg. cracks in reinforced concrete in Chao and Cheng (1998). However, DH is not yet capable of analysing together all the pathologies of a building for a full and auto-complementary diagnosis. Secondly, the aim of DH is quite different from the aim of most of the other similar expert systems because it is not oriented to achieve a full evaluation of the structural security of a building or to obtain a damage level, but to help a little-experienced human inspector in the first on field inspections at the detection of the most important and common pathologies in order to reach a most efficient resources distribution, assigning the high-experienced inspectors to the

unusual or most complex pathologies. However, in order to have some simple damage level indicator, DH displays a unique warning message if it diagnoses one of the most dangerous pathologies.

Comparing DH with other significant expert systems in the field of building inspection it is possible to obtain a better perspective of which are the achievements of the work herein presented. For example, MDDS (Masonry Damage Diagnostic System) from Van Balen (1996 and 2001) is an expert system aimed to carry out in situ inspections of masonry damage due to air pollution. Most of the information included in MDDS came from literature what also happens with DH but the main difference is how this knowledge is codified. As MDDS only uses decision tables and no production rules, the last decision step done in DH (usually a production rule) is not so easy in MDDS. This makes difficult to deal with uncertainty even in the simpler way used in DH. Comparing MDDS with DH other similarities are observed: both used an expert system shell to be developed, are goal oriented so only the necessary questions are asked and divide the variables between those that are fundamental to come up with a diagnostic and those complementary which are not even used in MDDS while in DH are considered in the last questions to give more confidence in the system output. After doing the tests with the developed expert system, two main advantages of MDDS are noted comparing with DH: the possibility of on-line actualisation and the recommendation of further tests to improve the diagnosis.

Other possible comparison that could be made is between DH and REPCON, which was developed by Moodi and Knapton (2003). REPCON is aimed for offering a repairing solution for damages in concrete elements. Indeed, its knowledge is structured in 13 decision tables and just one of them is dedicated to diagnosis while it is the full purpose of DH which also has a greater scope. The main difference between REPCON and other expert systems in the field of damage evaluation of buildings that have been studied is that this one is able to evolve with the knowledge of the users. To do that it first assign a confidence level to the user based on the answers the expert system asked they. Then, if it is proved the user is an expert it is allowed

to modify the data base. The confidence of the information is also evaluated taking into account only the source of it. On the contrary, DH has a static knowledge which offers more confidence in the obtained results but less upgrading velocity. However, in this branch of knowledge where almost every single pathology has already been studied it does not seem a big deal. Finally, whilst DH is clearly oriented to the formation of novel inspectors, REPCON has also this possibility but it is not directly aimed for it, resulting in a more complex system that covers a deeper analysis than DH in concrete elements.

Finally, a comparison between DH and the KBES (Knowledge Based Expert System) presented by Lu and Simmonds (1997) has been carried out to highlight the possibility of incorporating calculus results as an input data to the expert system and, more important, to remark the possibility of reaching a global evaluation of the whole building from the evaluation of each element. This is a clear advantage in front of DH that shows not enough efficiency at gather all the diagnostic to offer a global assessment. On the other hand, the way the uncertainty is handled in the KBES presented by Lu and Simmonds (1997) is not the most useful for teaching purposes as it requires the user to assign a confidence level to each observed symptom. Moreover, the aim of DH differs from the aim of KBES by Lu and Simmonds (1997) because the latter makes an assessment of the building but does not identify the causes of the damages.

To conclude with all these comparisons, one more time (see section 4.4) it is pointed out that the way DH deals with uncertainty is simpler than the common way other expert systems analysed by the authors (REPCON or KBES by Lu and Simmonds) do because, from the point of view of the developers of DH, in a first on field inspection it is not necessary to assign a probability percentage to a detected pathology and it is even less necessary to request a confidence level on the observation made when the user is being trained. It is also worth highlighting the way DH has its knowledge encoded using both production rules and decision tables together, which results in a more flexible system, with the variables clearly classified by their importance and allows the possibility of handling uncertainty in the way it has been done.

A sample of DH (Doctor House) can be freely distributed under request. For greater detail on the procedure followed and the results obtained, the reader is referred to Bernat (2007).

## **7. Conclusions**

Nowadays the industry is slowly changing from classical construction towards sustainable building. Therefore, most of the recently constructed building will have to be surveyed and/or repaired during the following years. The field of construction is clearly marked by an ever increasing amount of “illnesses”, but “doctors” are not being trained quickly enough to be able to address these cases with a level of quality required by the market. Hence, there is a pressing need for a commercially available tool for this area, which would not only increase productivity in technical inspections but would also allow for on-going training of the professionals who use it, as well as it could easily become a research and data compilation tool on the “health” of buildings.

After DH was developed and evaluated in the first phase, the following conclusions were reached:

- Expert systems were determined to be the best platform on which to develop software for decision-making on diagnosis of building pathologies, despite the difficulty in establishing internal order that is inherent to these types of programs.
- The type of damage that is easiest to classify is that affecting concrete structures, as they share many similarities, which limits casuistry and the data required for diagnosis. In contrast, the most difficult ones are those that affect masonry elements, which are more complex and are difficult to encode. Diagnosis of these pathologies requires diverse data. This means that the module rules for diagnosing masonry elements are more basic, less compact and have more entry objects.
- Treating uncertainty qualitatively is a feasible approach; it provides good results. Indeed, for building pathology diagnosis, this treatment is actually recommended, as it



provides greater transparency to the user and greater value in terms of critical analysis of the presented results.

- The interface could be easily improved and the knowledge base could be expanded in all areas of building pathology.

The last and most important conclusion is that the development of an expert system for diagnosing building pathology is feasible; indeed, creating a new, commercially available tool for practical applications is merely a question of providing the needed resources.

### **7.1. *Future work***

In 6<sup>th</sup> section some weaknesses of DH have arisen from analysing its results and comparing it with other expert systems which means some improvements should be done in the future. These future developing directions would include the following activities:

- to widely promote the use of DH among professional inspectors for detecting the main weaknesses and come up with a statistically more significant range of results,
- to spread the knowledge data base which should grow to cover all possible common pathologies in every structural element,
- to join the actual independent modulus in order to face a global assessment of the building instead of the evaluation of each structural element by their own,
- to include graphical information to assure a better understanding of the damages,
- to increase the functionality including suggestions about further tests or reparation techniques,
- to open a new development branch to create a professional version aimed at diagnosing the most complex pathologies that should include a quantitative approach in handling uncertainty,
- to include a function aimed for recollecting the data of the real uses in order to obtain new information about real damages and its frequency,

- to interconnect DH's users to create a network that would help further improvements and faster upgrading,
- to add an optional calculus module in the hypothetical professional version in order to obtain analytical evidences that could back the obtained solution,
- to move the expert system on other informatics support allowing it to be executed in nowadays smartphones and making it universally available.

It is thought all this future work is necessary and would be worthy once the technical feasibility of this kind of software has been proved.

### **Acknowledgements**

The authors gratefully acknowledge the support provided by company ATISAE, Project under contract MATE and Project under contract VALTEC of Generalitat de Catalunya. The first author also enjoys a Ph. D. grant offered by the Spanish Institution of Civil Engineers.

### **References**

Albert, K. and Wu, W. (1997). "A knowledge-based application in engineering", *Advances in Engineering Software*, Vol. 28, No.8, pp. 469-486.

Anumba, C. J. and Scout, D. (2001). "Intelligent pathological assessment of housing subsidence", *Structures & Buildings*, Vol. 146, No. 2, pp. 183-193.

Bernat, E. (2007). *Sistema Expert per a la diagnosi de patologies en elements estructurals*, Degree Thesis, ETSECCPB, Universitat Politècnica de Catalunya, Barcelona, Spain.

Brito, J. and Branco, F.A. (1997). "An expert system for concrete bridge management", *Engineering Structures*, Vol. 19, No. 7, pp. 519-526.

Calavera, J. (2005). *Patología de estructuras de hormigón armado y pretensado*, INTEMAC S.A., Madrid, Spain.

Chao, C.J. and Cheng, F.P. (1998). "Fuzzy Pattern Recognition Model for Diagnosing cracks in RC Structures", *Journal of computing in civil engineering*, Vol.12, No.2, pp. 111-119.

Cheng, C.T., Chunping, O. and Chau, K.W. (2002). "Combining a fuzzy optimal model with a genetic algorithm to solve multiobjective rainfall-runoff model calibration", *Journal of Hydrology*, Vol. 268, No. 1-4, pp. 72-86

Giarratano, J. and Riley, G. (1994). *Expert Systems. Principles and programming*, PWS Publishing Co., Boston, MA, USA.

González, M.P., and Zapico, J.L. (2007). "Seismic damage identification in buildings using neural networks", *Computers & Structures*, Vol.86, No. 3-5, pp. 416-426

Harmon, P. and King, D. (1985). *Expert Systems: Artificial Intelligence in Business*, John Wiley & Sons Inc., New York, USA.

Jiang, A.N., Wang, S.Y. and Tang, S.L. (2011). "Feedback analysis of tunnel construction using a hybrid arithmetic based on Support Vector Machine and Particle Swarm Optimisation", *Automation in Construction*, Vol.20, No. 4, pp. 482-489.

Kaetzel, L.J. and Clifton, J.R. (1995). "Expert/Knowledge based systems for materials in the construction industry: State-of-the-art report", *RILEM Journal of Materials and Structures*, Vol.28, No. 177, pp. 160-174.

Kaklauskas, A., Zavadskas, E. and Trinkunas, V. (2007). "A multiple criteria decision support on-line system for construction", *Engineering Applications of Artificial Intelligence*, Vol. 20, No. 2, pp. 163-175.

Kim, Y.M. and Kim, C.K. (2007). "Fuzzy set based crack diagnosis system for reinforced concrete structures". *Computers & Structures*, Vol. 85, No. 23-24, pp. 1828-1844.

Koppány, A. (2005). “Diagnostic tools and a new building defect registration system”, *10<sup>th</sup> DBMC International Conference On Durability of Building Materials and Components*, International Council for Research and Innovation in Building and Construction, ASTM and CSTB eds., Lyon, France, 17-20 April, pp. 1067-1074.

Kumaraswamy, M.M. and Dissanayaka, S.M. (2001). “Developing a decision support system for building project procurement”, *Building and Environment*, Vol. 36, No. 3, pp. 337-349.

Lee, J., Liu, K.F.R. and Chiang, W. (1999). “A Fuzzy Petri Net-Based Expert System and Its Application to Damage Assessment of Bridges”, *IEEE Transactions on systems, man, and cybernetics – Part B: Cybernetics*, Vol. 29, No. 3, pp. 350-369.

Lu, X. and Simmonds, S.H. (1997). “KBES for evaluating R.C. framed buildings using fuzzy sets”, *Automation in Construction*, Vol. 6, No. 2, pp. 121-137.

Moodi, F. and Knapton, J. (2003). “Research into a Management System for Diagnosis, Maintenance, and Repair of Concrete Structures”, *Journal of Construction engineering and management*, Vol. 129, No. 5, pp. 555-561.

Muñoz, M. (1988). *Conceptos y patología en la edificación*, M. Muñoz ed., Sevilla, Spain.

Murlidharan, T.L., Durgaprasad, J. and Appa Rao, T.V.S.R. (1997) “Knowledge-based expert system for damage assessment and vulnerability analysis of structures subjected to cyclones”, *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 72, No. November-December, pp. 479-491.

Muttil, N. and Chau, K.W. (2006). “Neural network and genetic programming for modelling coastal algal blooms”, *International Journal of Environment and Pollution*, Vol. 28, No. 3-4, pp. 223-238.

Najjaran, H., Sadiq, R. and Rajani, B. (2006). "Fuzzy Expert System to Assess Corrosion of Cast/Ductile Iron Pipes from Backfill Properties", *Computer-Aided Civil and Infrastructure Engineering*, Vol. 21, No. 1, pp. 67–77.

Peter, P.F.C. (1996). *An expert system for diagnosis of problems in reinforced concrete structures*, Minor Thesis, Royal Melbourne Institute of Technology, City Campus, Melbourne, Australia.

Poon, J. (2004). "Development of an expert system modelling the construction process", *Journal of Construction Research*, Vol. 5, No. 1, pp. 145-158.

Rodríguez, A. (2000). *Siniestros más frecuentes en la construcción de edificios*, MUSAAT, Madrid, Spain.

The Masonry Society Construction Practices Committee, eds., (2004). *Masonry Inspection Checklist*, Boulder, CO, USA.

Tiryaki, B. (2007). "Application of artificial neural networks for predicting", *Tunnelling and Underground Space Technology*, Vol. 23, No. 3, pp. 273-280.

Tanarslan, H.M., Secer, M. and Kumanlioglu, A. (2012). "An approach for estimating the capacity of RC beams strengthened in shear with FRP reinforcements using artificial neural networks", *Construction and Building Materials*, Vol. 30, No. May, pp. 556-568.

Van Balen, K. (1996). "Expert system for evaluation of deterioration of ancient brick masonry structures", *Science of the Total Environment*, Vol. 28, No. October, pp. 247-254.

Van Balen, K. (2001). "Learning from damage of masonry structures, expert systems can help!", *III International Seminar on historical constructions*, Lourenço, P.B. and Roca, P., eds., University of Minho, Guimaraes, Portugal, November, pp. 15-27.

Xie, J.X., Cheng, C.T., Chau, K.W. and Pei, Y.Z. (2006). "A hybrid adaptative time-delay neural network model for multi-step-ahead prediction of sunspot activity", *International Journal of Environment and Pollution*, Vol. 28, No. 3-4, pp. 364-381.

Yamazaki, Y., Uchiyama, Y., Ito, K., Akimoto, M. and Terada, N. (1990). "Expert Systems and integrated construction planning", *Habitat International*, Vol. 14, No. 2-3, pp. 95-106.

Yehia, S., Abudayyeh, O., Fazal, I and Randolph, D. (2008). "A decision support system for concrete bridge deck maintenance", *Advances in Engineering Software*, Vol. 39, No. 3, pp. 202-210.

Zhang, H. and Xing, F. (2010). "Fuzzy-multi-objective particle swarm optimization for time-cost-quality tradeoff in construction", *Automation in Construction*, Vol. 19, No. 8, pp. 1067-1075.

Zhao, Z. and Chen, C. (2002). "A fuzzy system for concrete bridge damage diagnosis", *Computers & Structures*, Vol.80, No. 7-8, pp. 629-641.

Zhao, M.Y., Cheng, C.T., Chau, K.W. and Li, G. (2006). "Multiple criteria data envelopment analysis for full ranking units associated to environmental impact assessment", *International Journal of Environment and Pollution*, Vol. 28, No. 3-4, pp. 448-464.

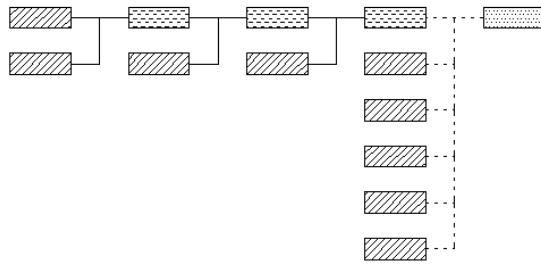


Figure 1. Encoding knowledge

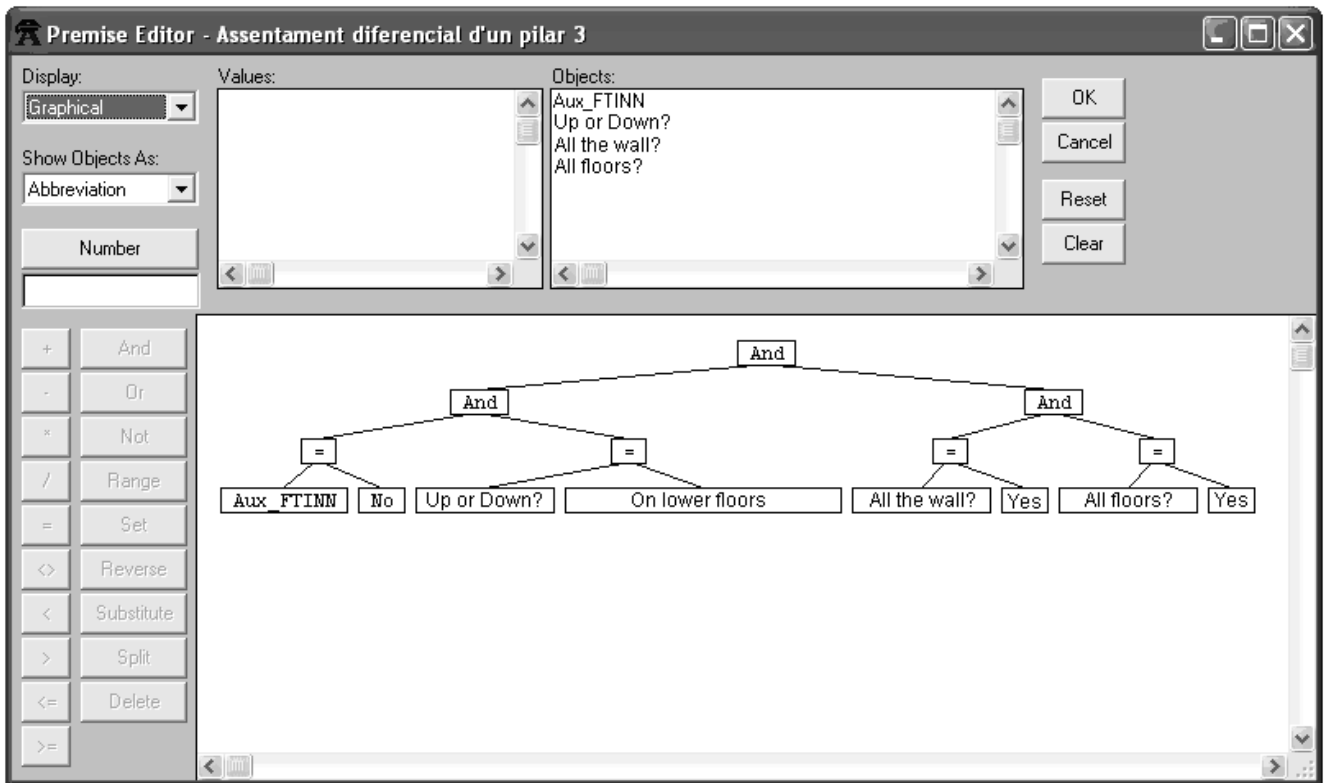


Figure 2. A production rule of the developed prototype

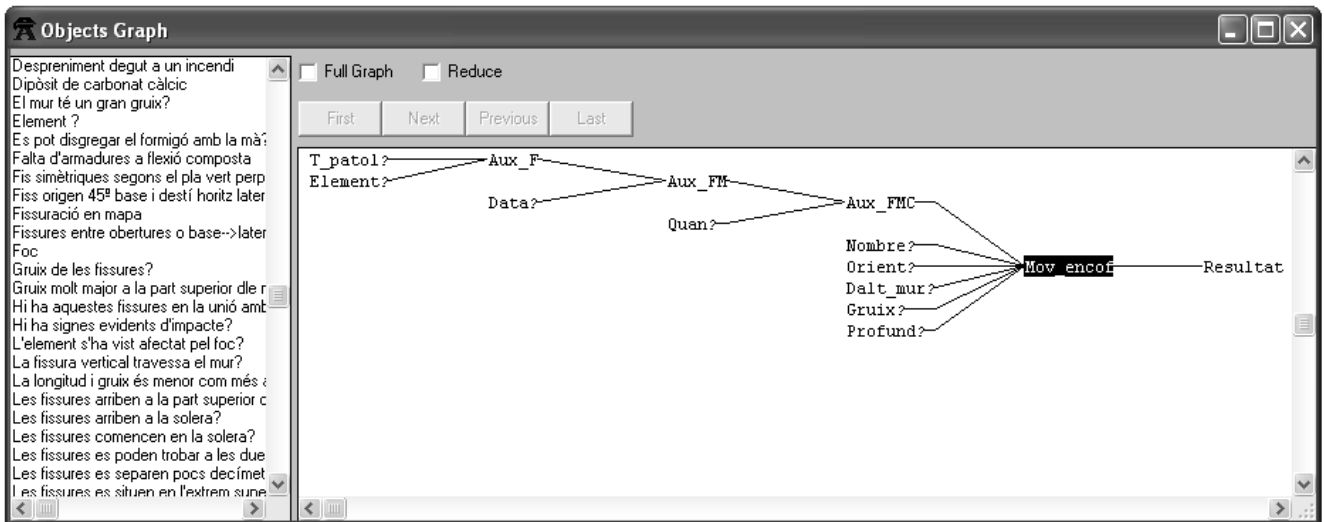


Figure 3. A pathology diagnostic tree

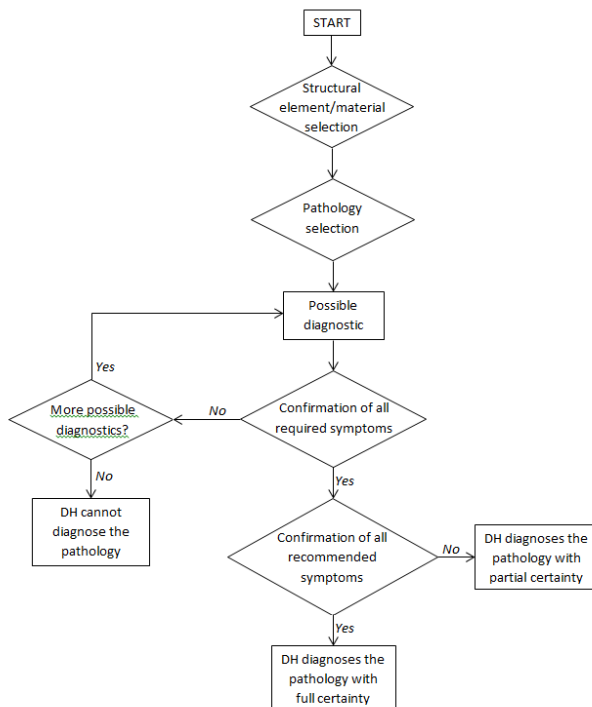


Figure 4. Flowchart of DH



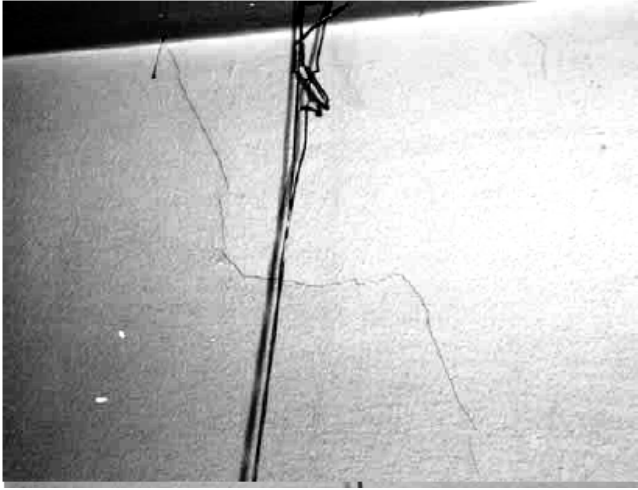


Figure 5. Crack to be analysed from side 1

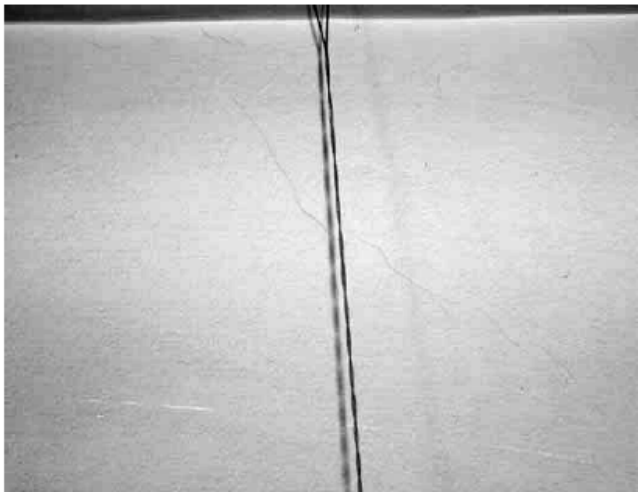


Figure 6. Crack to be analysed from side 2

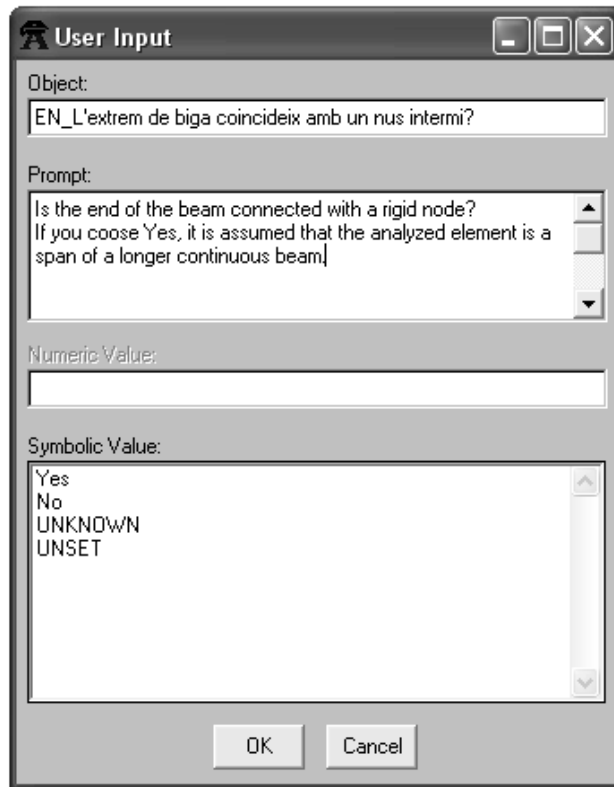


Figure 7. A screenshot of DH