Case-Based Reasoning and Fuzzy Logic in Fault Diagnosis

Viriato Marques[1,2], J.Torres Farinha[3], António Brito[4]
[1] GECAD – Knowledge Engineering and Decision Support Research Centre [2] Computer Science Department, [3,4] Mechanical Engineering Department
[1] ISEP – Engineering Institute - Oporto Polytechnic Institute [2,3] ISEC – Engineering Institute - Coimbra Polytechnic Institute [4] FEUP- Engineering Faculty of Oporto University
[1] Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto [2,3] Quinta da Nora, 3030-199 Coimbra
[4] R.Dr.Roberto Frias, 4200-065 Porto
PORTUGAL
[1] viriato@mail.isec.pt [2] tfarinha@mail.ipc.pt [3] acbrito@fe.up.pt

Abstract: - This paper is divided into four parts: the first one introduces SADEX, a fuzzy Case Based Reasoning (CBR) System for fault diagnosis. The second focus on its observation relevance factors and shows how the results are in complete agreement with the relevance concept introduced by Robertson and Spark-Jones in their well known and proved technique for document retrieval. The third describes how equipment composition information can be used to generalize and adapt case solutions to new and unknown occurrences; this generalization is based on a taxonomic similarity between functionally autonomous modules (FAMs). Finally the MKM - Maintenance Knowledge Manager system is introduced.

Key-Words: - Case-based Reasoning, Fuzzy Systems, Relevance, Taxonomies, Knowledge Management

1 Introduction

The CBR paradigm was introduced at the Yale University (U.S.A.) in 1983 (CYRUS system). At Europe it arrived about 1990 with PATDEX, a fault diagnosis system developed at KaisersLautern University (Germany) [1].

The operation cycle of a CBR system is well described by the Aamodt and Plaza diagram [2]. Basically it is composed of four phases named Retrieve, Revise, Reuse and Retain: *Retrieve* involves the search and selection of past cases, more or less similar to a new, *query* case. *Reuse* may imply some kind of adaptation so that a past solution may be applied to the present case. *Revise* makes the presentation of the (reused) solution and deals with its correctness or failure. *Retain* records present cases classified as relevant for solving future ones.

This paradigm is extremely well adapted to the solution of problems like learning from experience, keeping experience available as needed and quick know-how transfer.

However, CBR on its own is not enough, as technical staff know-how makes effective use of subjective experiences depending on visual inspection, noise, smell and even approximate measurement of some attribute values. The translation of this kind of information is possible by means of Fuzzy Sets and Fuzzy Logic [3,4].

SADEX is a CBR fault diagnosis system tailored to learn with, and assist, the technical staff in its daily unplanned maintenance tasks. The prototype has been validated in the health equipment field, where technical manuals and historical information are sometimes unavailable.

2 System Basic Structure

2.1 Attribute Handling with Fuzzy Sets

Equipment faults are described by observations. In SADEX the first element of an observation is the ID of the observed attribute. The system considers three attribute types: *Logical (L)*, (“On/Off”), *Measurable* (M) (“Temperature”) and *Non-Measurable Subjective* (NMS) (“Smell”). These three types give rise to two kinds of observations called *Absolute Semantics Observations* (ASOs) and *Differential Semantics Observations* (DSOs). Some examples follow:

1. It doesn’t work
2. Temperature is 10ºC
3. Temperature is low
4. Burned smell is evident

These examples show that abnormality can be expressed in different ways. And in fact maintenance teams, in their daily work, make effective use of these linguistic possibilities. By the other side, one of the important issues in the CBR paradigm is the global similarity computation between the query case and the past cases that takes place in the Retrieve phase.

There are many ways of evaluating this similarity but most part make use of attribute values. However, in order to allow cases to be compared, a single and normalized representation is necessary as, for instance, 10ºC can't be directly compared with "*low temperature*".

According to this we may consider that examples 3 and 4 make use of *differential symbols* such as "*low*" and "*evident*" to express a difference from an expected value. They contain a judgment of the observer too. Examples 1 and 2 make use of an *absolute symbol* as they translate a simple fact, contain no implicit judgment and so they don't express a deviation from a normal value. In the observation type perspective and according to this, we call DSO an observation that makes use of a differential symbol and ASO an observation that makes use of an absolute symbol.

In the attribute type perspective, example 1 makes use of a *Logical attribute* (L) as its value may vary between "*true*" and "*false*" only; examples 2 and 3 make use of a *Measurable attribute* (M) as its value can be measured by appropriate instrumentation; and example 4 makes use of a *Non-Measurable-Subjective attribute* (NMS) as the "*intensity of a smell*", for instance, is a personal and subjective experience. Finally, in the attribute value perspective, examples 1 and 2 are *precise* or *certain* and examples 3 and 4 are *uncertain*. To express precise values, *logical* or *numerical crisp values* are used; to express uncertain values, *fuzzy numbers*, *intervals* or *linguistic terms* are used, according to the nature of the uncertainty [3, 4, 5, 6].

Combining observation semantics, attribute type and value in a single and exhaustive classification scheme, after some simplifications it is possible to conclude that

1. The L attribute type allways leads to an ASO;
2. The NMS attribute type allways leads to a DSO;
3. The M attribute type leads to an ASO if the attribute value is numeric or to a DSO if it is a linguistic term.

So, it becomes possible to infer the observation type (ASO or DSO) from the attribute type. And, as we are looking for a single representation scheme, ASOs can then be recognized and transformed into DSOs or vice versa. There is an important reason to choose the former of these hypothesis: an ASO can always be reduced to a DSO but the contrary is possible only if some additional information about the attribute "*typical range*" is known. Some definitions follow:

1. The *Absolute Domain* or *Possible Range* (*PR*) of an attribute is the set of values considered of interest for abnormal facts representation. For an L attribute $PR=\{0,1\}$. For a M attribute $PR=[pr_l, pr_r]$ where pr_l and pr_r stand for *PR* left and right limits; NMS attributes always lead to DSO's and so PR is meaningless for them;
2. The *Absolute Typical Range* ($\tilde{ATR}$) of an attribute is a fuzzy set whose support is the set of values it can assume in *PR* in its "*normal*" state;
3. The *Differential Range* (*DR*) of an attribute (L, M or NMS) is a subinterval of [-1,1];
4. The *Differential Typical Range* ($\tilde{DTR}$) of an attribute is the fuzzy set that translates the linguistic term *Normal* in *DR*.

The goal is to transform the *PR* representation of an attribute value into its equivalent in the *DR* domain. Simultaneously the associated observation will turn from ASO into its equivalent DSO. We call this process *normalization*.

NMS attributes are automatically normalized as they have their linguistic terms defined in *DR*. Besides that, as the NMS type is always the support of a DSO, the original observation is already of this type. For L attributes normalization has to be done but, due to their simplicity, this is a trivial task. For M-type attributes there are 2 cases:

1. If they're represented by a linguistic term (DSO), normalization is already done as linguistic terms are defined in [-1,1] in order to express negative or positive deviations from the "normal" term;
2. If they're represented by a numeric value (ASO), normalization has to be done:

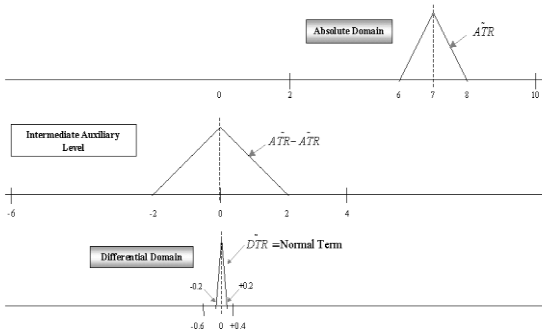

Fig. 1 - The Typical Range of a M-type attribute in the Absolute and Differential Domains

The steps of the normalization process are the same as for L-type attributes: let's suppose that we have *PR=[2,10]* and that $\tilde{ATR}$ is given by a fuzzy set in the α-cut notation such as (6,7,7,8). This fuzzy set, defined in *PR*, along with an auxiliary level and *DR* is shown in fig.1.

This figure also illustrates the important assumption that a correspondence has been established between $\tilde{ATR}$ and $\tilde{DTR}$: the fuzzy set $\tilde{ATR}$ corresponds to the linguistic term "normal" defined in $\tilde{DTR}$. This assumption has some consequences:

What values should define the $\tilde{DTR}$ term? First let's notice that this term is generated when, in the absolute domain, the attribute assumes the "normal value" $\tilde{v}$ or, $\tilde{v} = \tilde{ATR} = (a,b,c,d)_{ATR}$. In this case we'll get a difference $\Delta\tilde{ATR}$ given by:

$$\begin{aligned}\Delta\tilde{ATR} &= \tilde{ATR} - \tilde{ATR} = \\ &= (a,b,c,d)_{ATR} - (a,b,c,d)_{ATR} = \\ &= (a-d, b-c, c-b, d-a)_{ATR}\end{aligned} \tag{1}$$

This fuzzy interval is represented at the auxiliary intermediate level of fig.1. As *a-d* is always symmetrical of *d-a* and *b-c* of *c-b*, we verify that $\Delta\tilde{ATR}$ has always a zero medium value and is symmetrical, independently of the shape that $\tilde{ATR}$ may assume. Besides this, the maximum differences allowed between any possible attribute value and its $\tilde{ATR}$ are $pr_l - \tilde{ATR}$ for the left (lower) limit, and $pr_r - \tilde{ATR}$ for the right (upper) limit. So, the limits for the fuzzy set represented in the auxiliary level of fig. 1, α and β, are:

$$\alpha = \min\left(pr_l - \tilde{ATR}\right) = pr_l - d_{ATR} \tag{2A}$$

$$\beta = \max\left(pr_r - \tilde{ATR}\right) = pr_r - a_{ATR}\,. \tag{2B}$$

In our example we get α*=-6* and β*=4*. We have, then, an "intermediate domain" whose width, *w*, is given by

$$\begin{aligned}w &= \left|pr_l - d_{ATR} - (pr_r - a_{ATR})\right| = \\ &= (pr_r - pr_l) + (d_{ATR} - a_{ATR})\end{aligned} \tag{3}$$

So, *w* is the value by which we must divide $\Delta\tilde{ATR}$ in order to normalize it and obtain the "*normal term*" in the differential domain, $\tilde{DTR}$. Defining θ as:

$$\theta = (pr_r - pr_l) + (d_{ATR} - a_{ATR}) \tag{4}$$

then, combining eq. 1 and 4:

$$\begin{aligned}\tilde{DTR} &= \Delta\tilde{ATR}/\theta = (a,b,c,d)_{DTR} = \\ &= (a-d, b-c, c-b, d-a)_{ATR}/\theta\end{aligned} \tag{5}$$

This gives the *(-0.2, 0, 0, 0,2)* fuzzy interval that represents the "normal" linguistic term shown at the *differential domain* of fig.1.

This also implies that the limits of the differential domain are variable and depend on the location of $\tilde{ATR}$ in *PR*. In fact, let *DR=[*dr_l*,* dr_r*]*. Combining eq.2 and 4 we have:

$$dr_l = \alpha/\theta = (pr_l - d_{ATR})/\theta \tag{6A}$$

$$dr_r = \beta/\theta = (pr_r - a_{ATR})/\theta \tag{6B}$$

For our example we get dr_l*=-0.6* and dr_r*=0.4*, visible at the differential domain of fig.1. In other words, as the absolute typical range is more or less near one of the limits of *PR*, so the limits of the differential domain will differ from the central situation where they would be -0.5 and +0.5. In any case, the width of the differential domain is granted to be unitary.

Every time the system user makes an attribute definition by specifying some of the elements described in the previous section, an weighted mean of all its past definitions with the new one is made (for each similar equipment group). The results are sets of linguistic terms defined by fuzzy sets that attempt to catch the concept *linguistic terms usually used to describe abnormal attribute values for a given equipment group*. We call these sets of fuzzy sets *Conceptual Models* and this operation *Conceptualization.*

These *Conceptual Models* work as templates when a new definition is made. As the user can adapt their original values to a new definition we call this operation *Adaptation*.

Finally the *Projection* operation is symmetrical to *normalization* and simulates the reality view we have as modeled by previously defined concepts.

According to this, attribute representation and handling can be conveniently supported by the 3-Level Model (3L Model) shown in fig.2. This model is composed of three levels (External, Operational and Conceptual) related with the attribute domain and human perception:

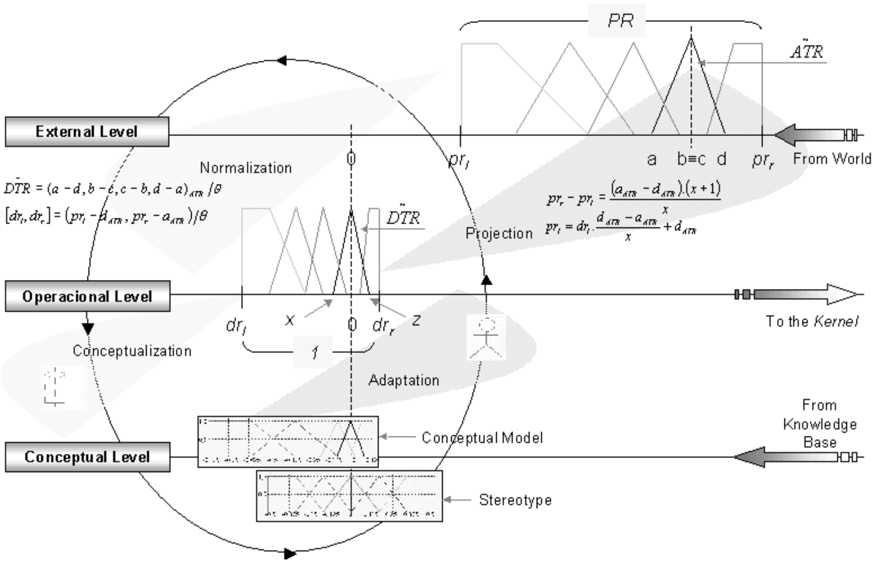

Fig. 2 - A 3-Level Model for attribute handling

1. External Level / Absolute Domain / Reality - models the reality, the real world;
2. Operational Level / Differential Domain / Human Cognition (reasoning) - handles normalized attribute values and supports every computation;
3. Conceptual Level / Conceptual Domain - / Human Cognition (learning) - handles meta-models for future attribute definitions.

2.2 Composition Information

The second element of an observation is the ID of the object that owns the attribute. This object can be the equipment - "the temperature of the oven is low"- or one of its components - "the relay of the thermostat of the oven is burned".

Equipment is classified into Homogeneous Equipment Groups (HEGs). An HEG is a leaf of a classification tree and may contain a set of equipment models. Each HEG has its own composition information. This means composition information is available for each HEG and not for equipment items. To achieve this, components are identified by their functional names, i.e. *transformer* or *step motor*, rather than by manufacturer ID or serial numbers. Each one of these elements is called a Functionally Autonomous Module (FAM). For our purposes this is enough, as we are interested in common faults and daily maintenance tasks, the kind of situation in which the goal is to locate the origin of the problem expressed in common language terms like t*he primary winding of the transformer of the high voltage power supply is burned*. This is accomplished by the so-called FAM-Chains as, for instance, *Primary Winding (Transformer (High Voltage Power Supply))*. A FAM-Chain is a series of ordered FAMs. A FAM-Chain doesn't include the HEG to which it belongs to: this means that the same FAM-Chain may belong to just one or to several HEGs (this is an important issue for generalization purposes). Each FAM is classified by means of a classification tree. This way relationships of the form *x Is-A y* are established between FAMs.

2.3 Observation Representation

An observation is the association of an attribute ID, an HEG or a FAM-Chain ID and a normalized attribute value. This association is formally supported by the Relational Assignment Equation (RAE) [3,4]. RAE basic form can be complemented with qualifiers ("*frequently*") and quantifiers ("*every*") that help to express occurrence frequency and spread.

Combining all the above referred components a Canonical Observation Form (COF) is defined as:

[Qf] (Attribute ([Qt] ([FAM-Chain](HEG)))) =
= [Log.Op.], [Mod.], Value, [Unit])

where *Qf* is a frequency qualifier, *Qt* is a quantifier, *Log.Op*. is the "Not" operator, *Mod.* is a modifier, *Value* is the normalized attribute value, *Unit.* is the measurement unit (for M type attributes) and *[..]* stands for "optional". Every COF is represented by the sextuple

$$(AtribID,\ FAM\text{-}ChainID,\ \mu_A(u),\ mu,\ \mu_F(v),\ \mu_Q(x))$$

where $\mu_A(u)$ is the membership function of the normalized fuzzy attribute, *mu* is the measurement unit, and $\mu_F(v)$ and $\mu_Q(x)$ are the membership functions of the frequency qualifier and quantifier, respectively.

3 Observation Relevance

3.1 Deriving Relevance Factors

Every fault has one or several causes. A cause may be a component malfunction (internal cause), an external factor or an operating mistake (human cause). A fault acts as a source of information in the sense that it produces messages sent from inside the equipment to the external world. These messages are abnormal attribute values. An observation takes place when one of these messages is perceived. Diagnosis is the task of identifying the causes of a fault, based on observations.

In a fault diagnosis CBR system, the similarity between a set of observations that describe the query case and sets of observations that describe past cases is evaluated. This way, the diagnosis associated with the most similar case can be pointed out as having the highest certainty factor, possibly being the correct one. This algorithm, Nearest-Neighbor, usually takes into account the observation relevance for each of the possible diagnosis under consideration, expressing it as a weighting-factor in the evaluation of similarity functions. What is relevance exactly?

As the right diagnosis is unknown *a priori*, then a relevance factor must be established for every pair observation / diagnosis-under-consideration. So, whatever it may be, it will have 2 dimensions: let's denote it by R_{ij}, where i is the observation O index and j the possible fault F index. As a starting point, we'll consider the limit situations we want relevance to describe:

1. R_{ij} should be 0 if fault F_j never generates observation O_i as in this case O_i nothing informs about its origin as it never happens. So, $P(Oi|Fj)=0$;
2. R_{ij} should be 0 if every possible fault can generate observation O_i as in this case O_i nothing informs about its origin;
3. R_{ij} should be 1 if just one kind of fault F_j can generate observation O_i and generates it always - because in this case fault F_j is uniquely *and* always identified by the occurrence of O_i.

According to 3., the second condition of the *and* can be translated by $P(O_i|F_j)$ as this probability will be 1 if F_j always generates O_i. The first condition captures the idea "how many distinct fault types can generate a given observation", that resembles restriction 2. This *and* condition also suggests that relevance should be expressed by a product of two terms: restriction 1 and the second part of restriction 3 suggest that one of these factors should be

$$\alpha_{ij}=P(Oi|Fj) \quad (7)$$

And the other one, β, is for the moment unknown. So

$$R_{ij} = \alpha_{ij}.\beta_i \quad (8)$$

where β_i must satisfy the following conditions:

A. Express the idea "how many distinct fault types can generate the same observation";
B. For a given observation it must be 0 if every fault can generate it;
C. For a given observation it must be 1 if just one fault can generate it.

Restriction A suggest expressing β_i as a quotient between $\#S_i$ - the number of distinct faults that can generate observation O_i - and the total number of faults that the system knows, #P:

$$\beta_i = \frac{\#S_i}{\#P} \quad (9)$$

Condition B is not satisfied as, if every fault type can generate observation O_i then $\beta_i=1$. In a heuristic perspective, a solution to turn the 1 into 0 would be the use of logarithms. But β_i still wouldn't satisfy condition C. However, the fact that "a fault acts as a source of information", as already referred, suggests Information Theory [7] as a working base: if an information source can generate k messages each one with a probability p_k, then the information contents I_k of each message is given by:

$$I_k = \log_2 \frac{1}{p_k} \text{ (bits)} \quad (10)$$

The first interesting thing about eq.10 is that it uses the *log* function already suggested. But besides that, eq.10 expresses a relative frequency, an approximation of the probability of observation O_i being generated by all the faults that can generate it if these faults occur with equal probability and always generate O_i. So, substituting the p_k of eq.10 by the β_i of eq.9, we'll obtain the information contents about how much observation O_i is discriminative of the faults that can generate it:

$$I_i = \log_2 \frac{1}{\frac{\#S_i}{\#P}} = \log_2(\#P) - \log_2(\#S_i) \quad (11)$$

Eq.11 satisfies condition B. However, condition C is still not satisfied as for $\#S_i=1$, $I_i \neq 1$ (in fact I_i will rise arbitrarily depending on the number of known faults, $\#P$). However, just dividing the second member of eq.11 by $\log_2(\#P)$ (and replacing the I_i by our factor

β_i,) C is satisfied as we get:

$$\beta_i = 1 - \frac{\log_2(\#S_i)}{\log_2(\#P)} \qquad (12)$$

3.2 Relevance in Document Retrieval

A well known and proven technique for document retrieval is described in [8]. According to the authors, "*the idea behind term weighting is selectivity: what makes a term a good one is whether it can pick any of the few relevant documents from the many non-relevant ones"*. The parallelisms between term / observation and document / fault are evident. Exploring further and referring to the definitions contained in [8], Term Frequency TF_{ij} is similar to α_{ij} (in fact TF_{ij} can be thought of as a particular case of α_{ij}) and Collection Frequency CFW_i is similar to β_i. Finally the authors of [8] note that *"a term that occurs the same number of times in a short document and in a long one is likely to be more valuable for the former"*. This assumption, that hasn't been considered yet, seems, however, very adequate: in fact, for the same observation a less prolific fault must assign it a greater relevance. In the extreme situation where a fault can generate just one kind of observation and another fault can generate this same observation plus, let's say, 4 or 5, then such observation must have higher relevance for the former fault. In this context [8] defines a Normalized Document Length, NDL_j. The application of this concept to the diagnosis domain generates, for each fault F_j, a factor that we call *Normalized Syndrome Length* NSL_j defined by:

$$NSL_j = \frac{M_j}{M} \qquad (13)$$

where M_j=average number of observations for all the occurrences of fault F_j and M=average number of observations for all faults.

Finally the SADEX expression for observation relevance becomes:

$$R_{ij} = \frac{\alpha_{ij} . \beta_i}{NSL_j} \qquad (14)$$

where α_{ij}, β_i and NSL_j are given by eq.7, 12 and 13 respectively.

3.3 Relevance Handling

Relevance factors are handled by matrixes that express, for each diagnosis under consideration, the relevance of each observation that, in the COF form, describes a fault.

Atrib	CMfa	Clas.	1	2	4	7					
5		-1			1.238						
5		1	0.385	0.385		0.145					
6		1				0.232					
6		2				0.232					
11		1			0.619						
16		-1				0.696					

Fig.3 - A relevance matrix

Relevance factors are updated according to eq.14 whenever a case library update occurs.

4 Induction and Case Adaptation

As described in section 1, the basic CBR mechanism looks for known cases whose description matches the one of the new case, so that a known solution can be proposed. However there is a high chance that a new case refers to HEGs or equipment type different from the ones the retrieved past cases contain, although some FAMs or FAM-Chains match or are somewhat similar, attribute ID being the same in new and past case(s). In these situations SADEX still looks for a solution making use of three generalization mechanisms:

1. As HEGs are sets of equipment models, every case about a given model that belongs to a given HEG can be used to every model that belongs to this HEG. So, a new case about X-Ray A can be used for X-Ray B as long as they belong to the same HEG;
2. A past case about HEG_1 can be used to solve a case about HEG_2 if two conditions hold: a) FAM-Chains of HEG_1 and HEG_2 to which current case COFs refer to must be the same; b) The diagnosis associated with HEG_1 must be possible to apply to HEG_2, i.e. they reference only FAM-Chains known as being part of HEG_2;
3. The third mechanism is used if both of the above fail. Now SADEX searches for HEGs such that condition 1 holds and condition 2 "almost" holds: instead of looking for exactly equal FAM-Chains in other HEGs, it looks for FAM-Chains that have a non-zero similarity with the ones of the new case.

The similarity, S, between two FAM-Chains is expressed as the product of the taxonomic similarity TS_i between each two correspondent FAMs of each chain:

$$S = TS_1 \times TS_2 \times ... \times TS_n = \prod_1^n TS_i \qquad (15)$$

So, if one FAM is "completely" different from its correspondent one, FAM-Chain similarity becomes zero. By the contrary, if it is the same FAM, the respective term will be 1.

Taxonomic similarity is a widely discussed issue [9,10,11,12]. Particularly [11] and [12] give precise methods to evaluate it. However, the similarity between a query object (i.e. the FAM of the current case FAM-Chain, *C*) and an object in the tree (i.e. the FAM of the past case FAM-Chain, *P*) implies having some knowledge about the attributes that these objects have in common. For our application the important attributes of such objects are the FAMs that compose them, as it is this feature that matters: for instance, the fact that a "*Regulated Power Supply*" and a "*Switching Power Supply*" use different devices to regulate their output voltage. However, this knowledge is usually unavailable.

This gives rise to various similarity evaluation methods, according to the object nature. References [11, 12] describe several cases. The following are important to our application (fig.2, 3 and 4):

1. C is a concrete object and P too. For instance, "*vertical potter*" and "*horizontal potter*";
2. C is a concrete object and P is a class. For instance, "*Switching Power Supply*" and "*Low Voltage Power Supply*";
3. C is a class and P is a concrete object. This is symmetrical to case 2.
4. C and P are both classes. For instance "*Video Monitor 9*" and "*Video Monitor 14*"

For situation 1, similarity is computed based on the specific attribute values of the objects. For 2, 3 and 4, three similarity definitions can be considered: a pessimistic, an optimistic and an average approach. For the average approach, the adaptation of the suggested expressions to our application gives, for 2, 3 and 4, respectively:

$$SIM(C,P) = \sum P(p).sim(c,p) \qquad (16)$$

$$SIM(C,P) = \sum P(c).sim(c,p) \qquad (17)$$

$$SIM(C,P) = \sum P(p).P(c).sim(c,p) \qquad (18)$$

where *P(p)* is the probability of an unknown attribute in *P* having an equal value to the corresponding one in *C*; *P(c)* is the probability of an unknown attribute in *C* having an equal value to the corresponding one in *P*; and *sim(c,p)* is the similarity between the incompletely known objects *C* and *P*, based on their common (and so known) attribute values.

We call *Intra-Class Similarity*, S_{Intra}, the similarity between children of the same node. But, as properties aren't defined for any object, the distance between objects can't be evaluated. So, all what's possible is to get an average distance (fig.4). In the example, PS is an abbreviation for "Power Supply" and PS1 to PS4 are 4 possible types of power supplies; *C* is a PS1 and *P* is a *PS2*. The distance between *C* and *P* is the average distance 1/4. For the general case, if the number of children is *N*, the normalized average distance is *1/N*. So, the intra-class similarity becomes:

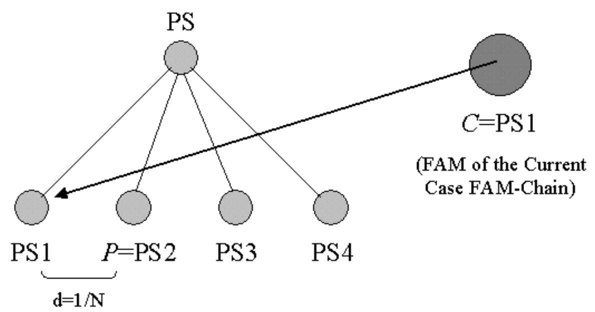

Fig. 4 - Similarity between objects, both instances of the same class

$$\begin{cases} S_{intra} = 1 \ \text{ if } P = C \\ S_{intra} = 1 - \dfrac{1}{N} \ \text{ if } P \neq C \end{cases} \qquad (19)$$

Inter-Class Similarity, S_{Inter}, is the similarity between any two nodes such that one of them is a child of the other one. *SIM(C,P)* is an Inter-Class similarity. Situation 2, for which *C* is a concrete object and *P* is a class, means that the FAM of the FAM-Chain under consideration for the current case is, for instance, *C=PS1*, and the FAM of the FAM-Chain for the past case under consideration is *P=PS* (fig. 5).

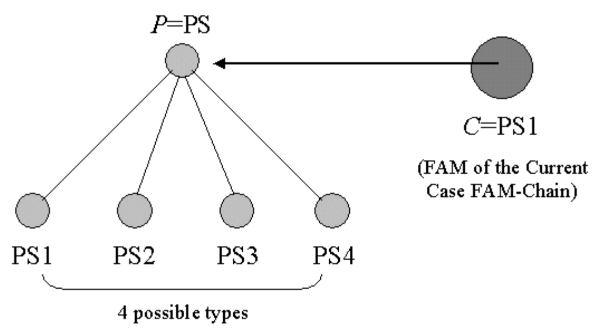

Fig. 5 - Similarity between object and class

According to eq.16, the probability of P=PS being a PS1, must be computed. For the example, this probability is $P(p)=1/4$. In the general case, $P(p)=1/N$ where N is the number of children of the class P. Then, the similarity between the common attributes in PS and PS1 must be computed. But here we have a problem as we don't deal with attributes.

To overpass this, we evaluate an average distance between C and P as the probability of C being equal to P times the distance between P and C (0 in this case), plus the probability of C being different from P, times the average distance between P and C. For the example we have:

$$d_{Inter} = \frac{1}{4}.0 + \frac{3}{4}.\frac{1}{4} = \frac{3}{16}$$

And for the general case this distance is:

$$d_{Inter} = \left(1 - \frac{1}{N}\right).d_{Intra} \qquad (20)$$

So, the Inter-Class Similarity becomes:

$$S_{Inter} = 1 - \left(1 - \frac{1}{N}\right).d_{Intra} \qquad (21)$$

Situation 3 is symmetrical to this one. Finally we consider situation 4, where C and P are both a class. For instance, C=PS and P=PS (fig.4). According to eq.18, the probability of PS being a PS1 should be computed and used twice for replacing $P(p)$ and $P(c)$ in eq.18, as we've concluded from situations 2 and 3 that $P(c)=P(p)$. However, as we aren't dealing with properties of classes, there's no reason to consider that P=PS and C=PS can be different: From the point of view of the knowledge base, they're both exactly the same "thing". In fact, if the current case FAM-Chain refers to a "Video Monitor" and the past case FAM-Chain under comparison also refers to a "Video Monitor", why should they have any probability of being different, from all that is known? So, if C is a class and P is a class, we'll assume that $S_{Inter}=1$. This is what happens in situation 1 for C=P. In other words, for our application, dealing with two equal concrete objects is the same as dealing with two equal abstract objects (classes).

Eq.19 and 21 are the basis for traversing a FAM classification tree between any two nodes. That is, wherever current case FAM and past case FAM may be located, we can always reach one starting from the other. As we walk along the tree paths, the similarity between them decreases.

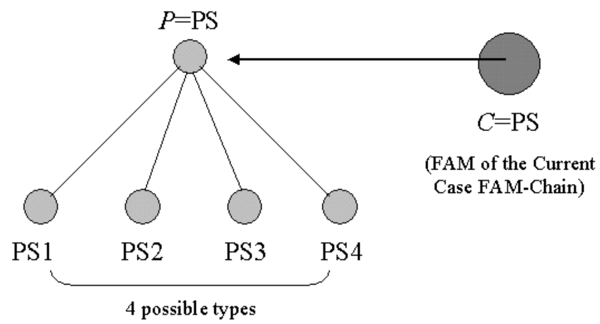

Fig. 6 - Similarity between class and class

It becomes evident that FAM classification must be carried out with some care and method. So, this process must be supervised by the System Administrator or any credited technician. As a rule of thumb we suggest that for each subdivision of a class, just one classification criterion be used. But all depends on common sense and practical situations.

5 The Diagnosis Process

The diagnosis process generates one positive and a maximum of three negative contributions for every diagnosis under consideration.

The positive contribution is generated by corresponding observations that both represent deviations from normal values (error signals) of the same sign, that is, both positive or negative. This feature implements *diagnosis selection by present symptom in the query case Q and the past case P_i*. This positive contribution can be emphasized or depreciated according to the value of its weighting factor w_{ij}.

The first negative contribution is also generated by corresponding observations that represent error signals of the same sign, but in which the attribute value in Q is somewhat *normal*. For this negative contribution a factor of the type *how much a value is normal* is computed according to the *fuzzy pattern matching* technique described in [13]. This feature implements *diagnosis exclusion by absent symptom in Q and present in P_i*.

The second negative contribution is generated by correspondent observations that represent error signals of contrary signs. This feature implements *diagnosis exclusion by present symptom in Q and absent in P_i* or vice-versa.

Absent descriptions in Q or in P_i are also taken into account and may lead to other negative contributions or determinate the beginning of an "ask-for-new-observation" cycle.

6 SADEX

The above exposed principles have been implemented in SADEX - A CBR Fuzzy System for Fault Diagnosis - tested in the health equipment maintenance field with very good results.

Availability through the Internet is an important issue as it allows a wide use of the system, also contributing to the enlargement of its knowledge base.

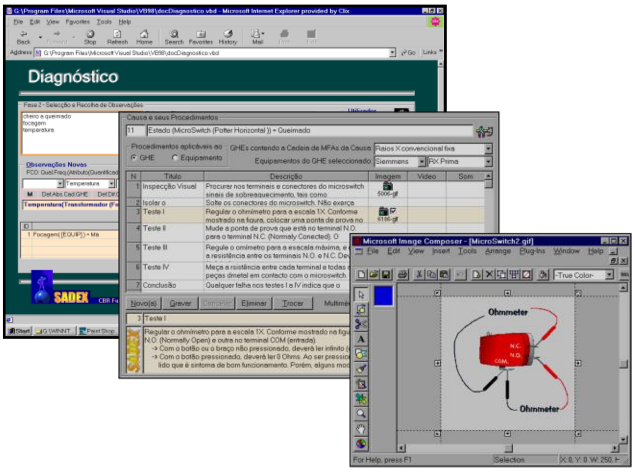

Figure 7 - Some forms of the system in operation

The software tools used in SADEX are standard, reliable and of low cost, allowing it to run on standard PCs. Fig.7 shows some forms of the system in operation.

7 MKM and Staff Training

Personnel's training and learning can be achieved through consultation and simulation modes. However, SADEX is part of a more general project that consists of an integrated health equipment maintenance system. Equipment and spare parts inventory control, planned maintenance procedures and information interchange between hospitals are important pieces of it.

But besides this, and as CBR based systems can been used for staff training purposes as they store organizational know-how in the form of past experiences, SADEX can be used as a know-how repository to be queried, when needed, for supplying examples in a learning context.

MKM - Maintenance Knowledge Manager - [21] is an intelligent e-learning platform tailored to maintenance staff training. This platform, under development, incorporates the above described CBR prototype as one of its major components, along with an agent based structure in the Internet environment [24]. Although tailored to the maintenance field, this system can also be used for teaching purposes in any other area.

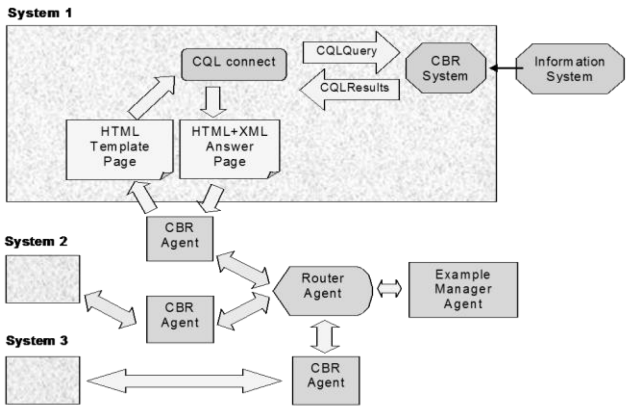

Figure 8 - Searching examples contained in CBR systems of distinct facilities

Fig.8 shows the purposed architecture for the CBR component of MKM. It contains three types of agents: CBR, Router and Example Manager. This architecture is derived from [14].

The Example Manager agent obtains the best examples from the connected CBR systems and chooses the ones that correctly illustrate the subject to teach.

The Router agent acts as a connector between all the CBR systems and the Example Manager, managing all the messages and their correct delivering.

The CBR Agent must query the CBR system it is responsible for, knowing some technical details about it. It also must collect the results and send them back to the Router Agent.

Incorporating an ITA - Intelligent Tutoring Agent - MKM can adapt itself to each student learning profile [15,16,17] as individual differences and profile definitions are of particular interest to providers of distance education systems where this phase plays a fundamental role in the success of the system [18,19].

8 Conclusion

In many domains past cases are available in the form of working orders, what allows a relatively easy and quick initial loading of CBR systems. Health equipment domain is no exception.

SADEX, a fuzzy CBR system for fault diagnosis, uses a general form of observation description that forces the user to describe cases in such a way that it becomes possible to infer the composition of HEGs and FAMs in terms of other FAMs. Each one of these modules has its own (observable) set of attributes whose values are commonly described by linguistic terms. This fact implies the manipulation of uncertain information using fuzzy sets and possibility theory.

Case retrieval and similarity computation uses taxonomic information about elements and distance between attributes, taking into account their relevance for the diagnosis under consideration. Relevance is updated whenever a case solution is confirmed, or a new case is added to the knowledge base.

The system allows Internet access and is part of a more general project in the field of Knowledge Management, a methodology that allows organizations to create extra value based on their own experiences, documentation, data, information and staff. CBR systems, by retaining daily acquired know-how expressed in an almost natural language form due to the support of fuzzy logic, can act as an excellent support for knowledge divulgation, implementing one of the phases that compose the so called "*knowledge management cycle*" [20]. Systems like SADEX [22] or SMITH [23] can provide the necessary cases that can be used as examples of the subject under study (Fig.8). This work is still going on.

Acknowledgements: The authors would like to thank the GECAD Research Unit for its support.

References:

[1] Richter,M., Wess,S., *Similarity, Uncertainty and Case-Based Reasoning in PATDEX*, Automated Reasoning - Essays in Honour of Woody Bledsoe, Kluwer Academic Press, 1997

[2] Aamodt,A., Plaza,E., *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*, AI-Com - Artificial Intelligence Communications, Vol7, 1994

[3] Zadeh,L.A., *Fuzzy Sets*, Selected Papers by Lofti A. Zadeh, 1965-1996, Advances in Fuzzy Systems - Applications and Theory, Vol.6, World Scientific, 1996

[4] Zimmerman ,H.J., *Fuzzy Set Theory and it's Applications*, Kluwer Academic Publishers, 1996

[5] Tsoukalas,L.H., Uhrig R.E., *Fuzzy and Neural Approaches in Engineering,* John Willey & Sons, 1996

[6] Ross,J.T, *Fuzzy Logic with Engineering Applications*, McGraw-Hill, 1997

[7] Shannon, *A Mathematical Theory of Communication*, The Bell System Technical Journal, 1948

[8] Robertson,S., Spark-Jones,K., *Simple, Proven Approaches to Text Retrieval*, Department of Information Science - City University - Cambridge - United Kingdom, 1997

[9] Osborne and Bridges, *A Case Base Similarity FrameWork*, University of York, United Kingdom, 1996

[10] Surma,J., *A Similarity Measure for Aggregation Taxonomies*, European Conference on Machine Learning ECML-97, 1997

[11] Bergmann,R., Stahl,A., *On the Use of Taxonomies for Representing Case Features and Local Similarity Measures*, 6th German Workshop on Case-Based Reasoning GWCBR'98, 1998

[12] Bergmann,R., *Similarity Measures for Object-Oriented Case Representations*, European Workshop on Case-Based Reasoning, EWCBR'98, 1998

[13] Cayrol,M., Farenny,H. and Prade,H., *Fuzzy Pattern Matching*, Kybernetes 11, 1982

[14] Jeon, Heecheol, Petrie, Charles, Cutkosky and Mark, *JATLite: A Java Agent Infrastructure with Message Routing*, Standford Center for design research (CDR), Standford, 2000

[15] Honey, P., Mumford, A., *The Manual of Learning Styles*, Maidenhead, Berkshire, 1992

[16] Mumford, A., *Effective Learning*, Institute of personnel and Development, London, 1995

[17] D. Kolb, *Experiential Learning: experience as the source of learning and development*, Prentice-Hall, New Jersey, 1984

[18] Aoki, K. and Bray, E., *Learning Styles of Distance Learners in Japan: Cultural Considerations*, 2007, http://aide.nime.ac.jp/research/Aoki%20&%20Bray.pdf

[19] Santally, M., *Learning and Perceptual Styles Consideration in the Design of Hypermedia Courseware for Distance Learning*, Institute of personnel and Development, Mauritius, 2003

[20] B. Newman and K. Conrad, *A Framework for Characterizing Knowledge Management Methods*, Practices and Technologies, George Washington Univ. Course EMGT 298, 1999

[21] Pincho,N., Marques,V, Farinha,J.T., Brito,A., *E-learning by experience - How CBR can help*, DIWEB'06 – 6th WSEAS Int. Conf. on Distance Learning and Web Engineering, Lisbon, Portugal, 2006

[22] Marques,V, Farinha,J.T., Brito,A., *SADEX – A Fuzzy CBR System for Fault Diagnosis*, FS'04 – 5th WSEAS International Conference on Fuzzy Systems, Udine, Italy, 2004

[23] Farinha,T., Fonseca,I., Marques,V., Brito,A., Marimba,A., Pincho,N., Simões,A., *A Global View of Maintenance Management – From Maintenance Diagnosis to Know-how Retention and Sharing*, WSEAS Conferences ASCOMS, TELEINFO, AEE, MCP and ICAP 2004, ISBN 960-8052-98-X. Cancun, Mexico, 2004

[24] Nern,J., Atasanova,T, Agre,G., Saarela,J., *INFRAWEBS - Intelligent Framework for Networked Businesses and Governments Using Semantic Web Services and Multi-Agent Systems*, 7th WSEAS International Conference on COMPUTERS, Corfu, Greece, 2003