CBR and MBR techniques: review for an application in the emergencies domain.

Carlos Mérida Campos
Emma Rollón

PURPOSE

The purpose of this document is to provide an in-depth analysis of current reasoning engine practice and the integration strategies of Case Based Reasoning and Model Based Reasoning that will be used in the design and development of the RIMSAT system.

RIMSAT (Remote Intelligent Management Support and Training) is a European Commission funded project designed to:

- Provide an innovative, 'intelligent', knowledge based solution aimed at improving the quality of critical decisions
- Enhance the competencies and responsiveness of individuals and organisations involved in highly complex, safety critical incidents - irrespective of their location.

The partners combine expertise in the management of complex scenarios, the management of knowledge, case based reasoning, model based reasoning, software and hardware development, training and commercial exploitation.

This document is part of a deliverable for RIMSAT project, and although it has been done in close contact with the requirements of the project, it provides an overview wide enough for providing a state of the art in integration strategies between CBR and MBR technologies.

The document is divided in three main chapters

In the first one we provide an in deep analysis of CBR and MBR paradigms. First we have focused on the foundations of CBR techniques, using the CBR paradigm architecture described by Aamodt & Plaza. The we explain the foundations of MBR. Given the wideness of the domain that the label MBR covers, our first task consisted of a survey about which techniques and reasoning approaches were the origins of what we call MBR. We also had to clarify what Model Based Reasoning is, as the limit between MBR and other similar techniques (as Rule Based Reasoning, Model Based Diagnosis, or simple deductive systems) is rather fuzzy. Moreover, the vagueness of language for defining concrete technical terms has caused several misunderstandings during the creation process of this document. The most important misconception we had was with topic *Model*. In RIMSAT we use two kinds of models: On the one hand, previously constructed models that describe certain procedures. These are use for learning about the domain where RIMSAT is placed and to construct a domain model in order to know which variables were relevant and which were superfluous. On the other hand we are dealing with models for doing MBR. Those models are the most relevant ones for this document. Some suggestions for naming both kind of models with different names were done, it was suggested the name paradigm and also pattern, but none of them are as accurate as model.

In this first chapter we review the most important CBR, MBR and CBR+MBR applications extracting the conceptual use of these techniques in the scenarios they are dealing with. A state of the art and review of existing applications and projects in Decision Support Systems based on CBR or MBR for emergency events constitutes the fourth section. The fifth section analyses the applicability of the techniques studied (CBR and MBR) within the framework of RIMSAT taking into account the user requirements. Finally in the last two subsections, a review and analysis of the previous analysis of the state of the art is undertaken, and a set of conclusions from the whole study is presented.

The second chapter analyses the integration issue from a more practical perspective. First, a concise introduction describes the integration levels. Following this, the second section uses as an input the conclusions of chapter 1 analysing the viability of each one (taking into account the concrete implemented components already described). The selected viable solutions for integration are depicted within the RIMSAT architecture in following section. Then, some possible models to reason with are identified and, finally, the analysis of this integration within RIMSAT.

Finally we provide a conclusions chapter were we summarise the most relevant founds of this research.

TABLE OF CONTENTS

1 MANAGEMENT SUMMARY

There are four parts in this document:

1. A study of the theoretical foundation of Case Based Reasoning (CBR) and Model Based Reasoning (MBR) techniques.
2. A description on how to integrate CBR and MBR techniques within RIMSAT with the existent components.

The research has focused on the clarification of what Model Based Reasoning is, as the limit between MBR and other similar techniques (as Rule Based Reasoning, Model Based Diagnosis, or simple deductive systems) is rather fuzzy. Moreover, the vagueness of language for defining concrete technical terms has caused several misunderstandings during the creation process of this document. The most important misconception we had was with topic *Model*. In RIMSAT we use two kinds of models: On the one hand, previously constructed models that describe certain procedures. These are used for learning about the domain where RIMSAT is placed and to construct a domain model in order to know which variables were relevant and which were superfluous. On the other hand, we are dealing with models for doing MBR. Those models are the most relevant ones for this document. Some suggestions for naming both kinds of models with different names were done. It was suggested the name paradigm and also pattern, but none of them are as accurate as model.

Regarding the definition of a hybrid system, four levels of integration have been specified between CBR and MBR in RIMSAT:

- The first consists simply of keeping both technologies separate and letting the user organisation (system administrator) choose one or the other approach depending on the circumstances. This toolbox approach should not be rejected because such a user may need only MBR or CBR.
- In the second level of integration, called the co-operative level, the technologies are kept separated but they collaborate. Each uses the results of the other to improve or speed up its own results, or both methods are used simultaneously to reinforce the results. For instance MBR can be used as an input for a case retrieved from the whole process of the CBR engine.
- The third level of integration, called the workbench level, goes a step further. The technologies are separated but a ‘pipeline’ communication, which is used to exchange the results of individual modules of each technology.
- The final level (the seamless level) aims at reusing the best components of each method to build a powerful integrated tool, which avoids the weaknesses of each separated technology and preserves their advantages.

Studying the state of the art of CBR and MBR architectures it has been identified many options that have been categorised into twelve generic possibilities (section 2.5.4). From them, and after the study of their applicability in RIMSAT, the following conclusions are reached:

- The concrete MBR techniques found in the literature are not directly applicable to model RIMSAT scenario and provide an accurate and useful output given the wideness of it. However, they can be fruitful for solving specific problems of the whole domain.
- The local similarity functions that are currently being used in the CBR systems are not optimal for our domain. We are dealing with a domain model whose variables are very co-related, and their inter-dependences are well understood. Such an scenario has suggested us the idea of creating a model with the expressivity of a Bayesian network in order to capture the well known

but complex network of dependences between the attributes and using this model for creating a new concept of local similarity weighting function.

- CBR and MBR can be used together at different levels of integration in RIMSAT domain.
- The role of CBR within RIMSAT has been always quite clear. Having a case base for managing the knowledge needed for facing certain situations. Conceptually, the kind of tasks that MBR could perform in our environment can be:
  - Using MBR for manipulating cases in the case base (factoring cases, correcting cases, branching cases...).
  - Using MBR for temporal projection (prediction).
  - Using MBR for revising alternative solutions
  - Using MBR for solution refinement or upgrade.
  - Using MBR for similarity assessment.
  - Using MBR for information retrieval.

The use of one or another will depend on the available information for implementing the models.

2 SURVEY OF CBR & MBR FOR RIMSAT

The University of Girona, in conjunction with Teradyne, Kaidara and Nemesia undertook an in-depth survey and analysis of case-based and model-based reasoning technologies and application areas to understand how these technologies could be used to provide a rapid-response, decision-support solution for incident and event management in emergency situations.

This chapter is structured in the following way: first CBR is studied; its principles, theoretical foundations, and a deep analysis of the similarity assessment is undertook. After CBR, MBR is presented; first its origins and basic theoretical foundations, then a deep study of the concept Model specifying the characteristics of models used for reasoning, what can we model and how can we do it. We also studied what are the modelling issues. Once models have been presented, we explain what are the Model Based Reasoning principles and the most typical architectures. After MBR subsection, we will find a review of applications. Finally, as a conclusion of the survey, there is a subsection about applicability of CBR and MBR in RIMSAT, where we find first of all a review of advantages and drawbacks for both systems, and finally the set of conclusions based on the study performed.

2.1 CBR BACKGROUND

2.1.1 Introduction

Case-Based Reasoning (CBR) helps to solve a current problem by retrieving the solution to previous similar problems and adapting those solutions to meet the current need. It is based upon previous experiences and patterns of previous experiences. Human-beings, with years of experience in a particular job and activity, use this technique to solve many of their problems (e.g. a skilled paramedic arriving at the scene of an accident will often know automatically the best procedure(s) to deal with a patient). One big advantage of CBR is that inexperienced people can draw on the knowledge of experienced colleagues, including those outside the organisation, to solve their problems.

CBR is a problem-solving paradigm that in many respects is fundamentally different from other major Artificial Intelligence (AI) approaches. Instead of relying solely on general knowledge of a problem domain, or making associations along generalised relationships between problem descriptors and conclusions, CBR is able to use the specific knowledge of previously experienced, concrete problem situations (cases). A second important aspect is that CBR is also an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for solving future problems.

In CBR terminology, a *case* usually denotes a problem situation. A previously experienced situation, which has been captured and learned in a way that it can be reused for solving future problems. A 'new case' or 'unsolved case' is the description of a new problem to be solved. CBR is a cyclic and integrated process of solving a problem, learning from this experience, solving a new problem, etc.

A very important feature of CBR is its coupling to learning. The notion of CBR does not only denote a particular reasoning method, irrespective of how the cases are acquired, it also denotes a machine learning paradigm that enables sustained learning by updating the case base after a problem has been solved. Learning in CBR occurs as a natural by-product of problem solving. When a problem is successfully solved, the experience is retained in order to solve similar problems in the future. When an attempt to solve a problem fails, the reason for the failure is identified and remembered in order to avoid the same mistake in the future. Effective learning in CBR requires a well worked out set of methods in order to extract relevant knowledge from the experience, integrate a case into an existing knowledge structure, and index the case for later matching with similar cases.

2.1.2 Principles of Case-based Reasoning

2.1.2.1 <u>The cognitive model</u>

Case-Based Reasoning (CBR) [Aamodt and Plaza, 1994; Kolodner, 1993; Riesbeck and Schank, 1989] derives from a view of understanding problem solving as an explanation process. The foundations of CBR rely on the early work done by Schank and Abelson [Schank and Abelson, 1977] where they proposed that our general knowledge about situations is recorded as *scripts*. The cognitive model behind the CBR is based on the theory of Dynamic Memory [Schank, 1982] that introduces indexing as the key to using experience in understanding. The main premise being that remembering, understanding, experiencing, and learning cannot be separated from each other, and that the human memory is dynamic, and changes as a result of its experiences.

CBR systems improve their performance, becoming more efficient, by remembering previous solutions given to similar problems and adapting them to fit a new problem, rather than having to solve it from scratch. This augments the idea of 'components of expertise' [Steels, 1990] using the solved cases as an episodic memory: the memorisation of problem-solved episodes allows methods to be integrated because they require accessing the past experience to improve the system performance. Furthermore, case-based reasoners become more competent in their evolution over time, so that they can offer better advice and guidance when facing less common situations, preventing them from repeating the same mistake(s) in the future (learning process).

2.1.3 Organisation and Representation of Cases

The reasoning by analogy of CBR involves collecting a considerable number of relevant cases, or experiences, in a particular domain. Storing a case means keeping a description of each experience as well as the solution provided to that experience. The set of stored cases or experiences is usually called the Case Library, the Case Base or the Case Memory. In the next subsections, the main Case Library organisations and case structures will be described.

2.1.3.1 <u>Case Library</u>

Main memory organisations in CBR systems can be summarised in two general approaches: 'flat memories' and 'hierarchical memories' such as shared feature networks, prioritised discrimination networks/trees or redundant discrimination networks/trees.

Flat memories always retrieve the set of cases that best match the input case. Moreover, adding new cases to memory is not time-consuming. However, flat memories have a major disadvantage in that the retrieval time is very time-consuming because every case in the memory is matched against the current case - commonly using a 'Nearest Neighbour' (NN) algorithm [Watson, 1996].

Alternatively there are hierarchical memories. In these kind of memories, matching process and retrieval time are more efficient, due to the fact that only a few cases are considered for similarity assessment purposes, after a prior discriminating search in the hierarchical structure. However, hierarchical memories also have some important disadvantages: keeping the hierarchical structure in optimal conditions requires an overhead on the Case Library organisation; the retrieval process could miss some optimal cases because it is searching in a wrong area of the hierarchical memory. This latter problem is particularly prevalent in prioritised discrimination networks/trees.

2.1.3.2 <u>Case Structure</u>

The cases stored in the Case Library are past experiences, which have been 'captured' and 'learned' in such a way that they can be re-used to solve future causalities. A case incorporates a set of features such as:

- an *identifier* of the case
- the *description* of the case
- the *diagnostic* of the case
- the *solution* of the case;
- the *derivation* of the case, i.e. from where the case has been derived/adapted;
- the *solution result* - information indicating whether the proposed case solution has been a successful one or not;
- an *utility measure* of the case in solving past cases when it was used;
- other *relevant information* about the case.

The following is an example of a case representation:

```
( :identifier  CASE-18
  :situation-description  ( (Water-inflow 35,198 m3/day)
                          (Inflow-Chemical-Oxygen-Demand 289 mg/L)
                                      . . . )
  :diagnostics  NORMAL-SITUATION
  :actuation-plan  ( (1 Maintain-the-numerical-control-algorithm)
                   (2 Adjust-Dissolved Oxygen (DO)-value)
                      . . . )
  :case-derivation  INITIAL-CASE / CASE-13
  :solution-result  SUCCESS /FAILURE
  :utility-measure  0.7
  :distance-to-case  0.3782 )
```

Figure 1.1. Example of a case representation

2.1.4 The Case-Based Reasoning Cycle

The basic reasoning cycle of a CBR agent can be summarised by a schematic cycle (see Figure 1.2 below). Aamodt and Plaza [Aamodt & Plaza, 1994] adopt the '4 Rs' schema:

- *Retrieve* the most similar case(s) to the new case.
- *Reuse* or *Adapt* the information and knowledge in that case to solve the new case. The selected best case has to be adapted when it does not match perfectly the new case.
- *Revise* or *Evaluate* of the proposed solution. A CBR agent usually requires some feedback to know what is going right and what is going wrong. Usually, it is performed by simulation or by asking to a human oracle.
- *Retain* or *Learn* the parts of this experience likely to be useful for future problem solving. The CBR agent can learn both from successful solutions and from failed ones (repair).

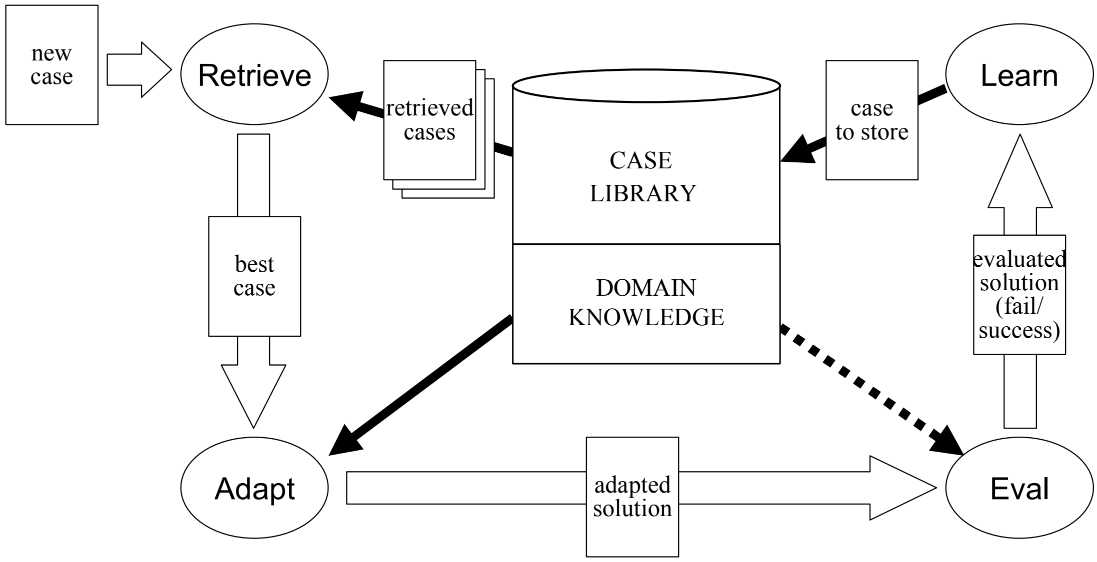


Figure 1.2. The general case-based reasoning paradigm

The quality of the new case(s) extracted by the CBR agent depends upon some of the following criteria:

a) The usefulness of the case(s) extracted and selected;

b) The ease of use this (these) case(s);

c) The validity of the reasoning process;

d) The improvement of knowledge through experience.

When setting a CBR agent one has to take into account some design decisions or static problems such as:

a) How to describe the domain problems?

b) Which will be the case structure?

c) Which will be the Case Library structure?

d) How to deal with the missing information problem?

e) Which will be the criterion for indexing the Case Library?

f) How to assess similarity between cases?

2.1.4.1 Case Retrieval and Similarity Assessment

The task of retrieving cases in the Case Library is slightly more difficult than typical retrieval in databases. In database systems the recalling algorithms use an exactly matching method, whereas in a Case Library retrieval, because the very nature of the structure, a partial-matching strategy is used. A retrieval method should try to maximise the similarity between the actual case and the retrieved one(s). Most of the time, this task implies the use of general domain knowledge.

The retrieving process of a case (or a set of cases) from the system's memory is highly dependent on the organisation of the Case Library. Major Case Library structures are flat memories or hierarchical ones. Flat memories have an intrinsic retrieval problem of poor time performance, so that retrieval time is proportional to the size of the Case Library - the larger the library, the slower the retrieval. Hierarchical memories are very efficient in time retrieval because only a few cases are considered for similarity assessment purposes, after a prior discriminating search in the hierarchical structure.

The retrieval process in case libraries usually consists of two main sub steps:

- *Searching* the most similar cases to the new case: the goal of this stage is recalling the most promising cases - given that the subsystem has a goal and therefore the relevance of the cases depends upon that goal - based on using some direct or derived features of the new case as indexes into the Case Library.
- *Selecting* the best case(s) from those collected in the previous step. Commonly this selection is made by means of a case ranking process through a similarity or distance function. The best-retrieved case is the closest one (most similar) to the new case.

Selecting the best similar case(s), it is usually performed in most CBR agents by means of some evaluation heuristic functions or distances, possibly domain dependent. They are usually named as nearest neighbour (NN or k-NN) algorithms [Watson, 1996]. The evaluation function usually combines all the partial matching through a dimension or attribute of the cases, into an aggregate or full-dimensional partial-matching between the searched cases and the new case. Commonly, each attribute or dimension of a case has a determined importance value (weight, normally scaled in the range 0..1 or equivalent), which is incorporated in the evaluation function. This weight could be static or dynamic depending on the purpose of the CBR agent. The evaluation function computes an absolute match score (a numeric value), although a relative match score between the set of retrieved cases and the new case can also be computed.

The properties of a similarity function should be the following:

- It should be reflexive - a case is always similar to itself.
- It should be symmetric - if case A is similar to case B, them case B is also similar to case A. In that sense, 'similarity' is related to the notion of 'distance'.

Finally, similarity is not always transitive. That is, if A is similar to B and B is similar to C, we cannot always conclude that A is similar to C. For example, a white BMW is similar to a white Renault and a white Renault is similar to a red Renault, but the white BMW is not similar to red Renault.

2.1.4.1.1 Feature Weighting techniques

Before describing the similarity measures, we have to cope with a related issue, that is, how to find out which are the relevant features to be taken into account. When experts are available in a particular domain, theses tasks could be easier. But, in general, when there is no expertise available, some automatic methods should be used. Many methods have been proposed and used in the literature for feature selection. One of this kind of techniques is *feature weighting*. Feature weighting consists in the assignment of an importance degree to each one of the available features describing a domain or process. Normally weights are scaled in the range 0..1 or equivalent. Thus, features with lower weights are the less important ones, while high weights mean very important features.

In later years, many researchers are focusing on feature weighting. As said before, feature weighting is a very important issue. It is intended to give more relevance to those features detected as important, and at the same time, it is intended to give lower importance to irrelevant features. Most general methods for feature weighting use a global scheme. It means to associate a weight to all the space of the feature. If a continuous attribute is present, a discretization pre-process is suggested to allow make a weight calculation according to his interval values and his correlation with the value class. The importance of one feature will be determined by the distribution of the class values for that feature. Some work has been done such as the mutual information technique proposed in [Wettschereck et al, 1997], the work reported by Mohri and Tanaka [Mohri-Tanaka, 1994] of his QM2 method and Creecy *et al.* [Creecy et al, 1992], and other work in [Kohavi et al]. On the other hand, local weighting methods assign specific weights to specific values of the attribute. Some works have been done such as the value difference metric of Stanfill and Waltz [Stanfill-Waltz, 1986], the class distribution weighting method of Howe and Cardie [Howe-Cardie, 1997], and the per-category feature importance criterion of Creecy et al. [Creecy et al, 1992].

In [Wettschereck et al, 1997] a five-dimension framework for feature weighting is presented: bias, weight space, representation, generality and knowledge. Taking into account that point of view, our approach can be classified as Bias: Preset, Weight Space: Continuous, Representation: Given, Generality: Global and Knowledge: Poor.

2.1.4.1.1.1 GLOBAL WEIGHTING TECHNIQUES

Global weighting algorithms compute a single weight vector for all cases. A weight is associated to each attribute and this weight is fixed for all the attribute space. There are some research works. Creecy et al. in [Creecy et al, 1992] introduced *cross-category feature importance* (CCF) method. CCF averages weights across classes using:

$$w(f) = \sum_{c_j \in J} p(c_j | f)^2$$

Where:

$P(c_j|f)$ is the conditional probability that an instance is member of c_j given its value for *f.*

They are trying to assign higher weights to features that occurred in fewer classes. However, this algorithm is not sensitive to the distribution of feature values across classes. Conditional probabilities have been used to assigning feature weights. Mohri and Tanaka [Mohri-Tanaka, 1994] reported good performance of his QM2 method that use a set of transformed features based on the originals. The Mutual Information (MI) method assign low weights to features that provides little information for classification, and higher weights to features that provide more reliable information. The mutual information is computed as follows [Wettschereck et al, 1997]:

$$w(f) = \sum_{v \in V} \sum_{c_j \in J} p(c_j, x_f = v) * \log\left(\frac{p(c_j, x_f = v)}{p(c_j) * p(x_f = v)} \right)$$

where $p(c_j)$ is the frequency of class c_j among the training set X and $p(x_f=v)$ is the frequency of value *v* for *f* among instances in X.

2.1.4.1.1.2 LOCAL WEIGHTING TECHNIQUES

Local weighting methods assign specific weights to specific values of the attribute. Is it possible that one attribute could be very useful predicting a class according to one of its values, but, when it takes another value, this attribute is not relevant. The value difference metric of Stanfill and Waltz [Stanfill-Watz, 1986], assigns a different weight to each value of the feature. In VDM the distance between cases is defined as follows:

$$distance(u, v) = \sum_{i=1}^{n} w(i, u_i) * \delta(i, u_i, v_i)$$

where:

$$w(i, p) = \sqrt{\sum_{c=1}^{C} \left(\frac{C_i(p, c)}{C_i(p)} \right)^2}$$

$$\delta(i, p, q) = \sum_{i=1}^{C} \left(\frac{C_i(p, c)}{C_i(p)} - \frac{C_i(q, c)}{C_i(q)} \right)^2$$

where *p* and *q* are possible values of an attribute, $C_i(p)$ is the total number of times that value *p* occurred at an attribute *i*, and $C_i(p,c)$ is the frequency that *p* was classified into the class *c* at an attribute *i*.

Howe and Cardie [Howe-Cardie, 1997] propose a class distribution weighting method, which computes a different weight vector for each class in the set of training cases using statistical properties of that subset of data. Creecy et al. [Creecy et al, 1992] use per-category feature importance to assign high weights to features that are highly correlated with the class.

2.1.4.1.2 Local similarities

From a higher level perspective, the overall evaluation of the similarity between two cases can be based on the computation of local similarities between each attribute. The local similarity may vary, depending on the attributes' type or the size of the sets on which the similarity is computed. For instance, 10 is more similar to 20 if the size of the possible interval varies from 0 to 1000, than if it varies from 0 to 20.

Local similarities are generally defined on a restricted interval, for example [0,1]. This normalisation enables the user to combine them to evaluate global similarities (see next section). Different local similarity measures can be used to cope with various data types. Theses measures are pre-defined in most of the CBR tools. However, it may be useful to define new local similarity measures better suited to a specific domain. Some tools allow the user to define them through a programming language, or even to set up the similarity matrix directly between attribute values.

2.1.4.1.3 Global similarities

Once a set of local similarities has been defined for each attribute, it is necessary to combine them in a global similarity measure. Hence, a global similarity SIM between two cases A and B described by *p* attributes, can be expressed by:

$$\text{SIM}(A,B) = F(\text{Sim}_1(a_1, b_1), \text{Sim}(a_2, b_2), \ldots, \text{Sim}_p(a_p, b_p)) \qquad (1)$$

Where $F: [0, 1]^p \rightarrow [0,1]$

Global similarity measures are usually pre-defined in CBR tools. For instance, the default global similarity in REMIND is the Block-City. It is important to have the possibility of providing a weight matrix in which the values can be freely defined by the user, or automatically computed by the tool. In **S^3-CASE**, the weight matrix computation can be achieved via a learning process.

Similarity assessment is a key part of reasoning with cases. However, no similarity measure is ever perfectly appropriate for all application domains. The best procedure is first to try the known similarity measures and then, if the results are not convincing, switch to more complex functions that are domain dependent. The determination of weights is also an important means of tuning a system where statistics knowledge acquisition and test can help achieve better results. It is therefore important to be able to define one's own similarity functions within the tools. Some tools offer the possibility of programming custom similarity measures.

Most case-based reasoners such as REMIND [Cognitive, 1992], MEDIATOR [Kolodner and Simpson, 1989], PERSUADER [Sycara, 1987] , etc., use a generalised weighted distance function (NN) such as:

$$dist(C_i, C_j) = \sum_{k=1}^{n} wk \; x \; atr\_dist(C_{ik}, C_{jk})$$

Some others measures have been defined such as in [Leake et al., 1997; Osborne and Bridge, 1998; Sànchez-Marrè *et al*., 1998; Warren *et al*., 1998].

Two kinds of impasse could happen in a CBR agent:

- The situation is unknown, i.e. there is no memory about this situation or there is not a successful solution to this situation
- There are several ways (solutions) to proceed, i.e. there are several methods that may be applicable to a situation with the same degree of confidence.

For these impasse situations, CBR agents can use the expert general knowledge coded into the system or can generate an alarm that has to be solved by the user. Other approaches such as in NOOS [Arcos & Plaza, 1995] generate a reflexive task whose goal is to solve that impasse.

2.1.4.1.4 Similarity measures

Currently, there are several similarity measures that have been used in CBR systems, and some comparison studies exist among these similarity measures (see [Wilson-Martínez, 1997] and [Liao-Zhang, 1998]). The results obtained in these studies show that the different similarity measures have a performance strongly related to the type of attributes representing the case and to the importance of each attribute. Thus, is very different to deal with only lineal or quantitative data (continuous), with discrete or qualitative (entire) or nominal (qualitative not ordered). To give a greater distance contribution to a more important attribute than other less important attributes is necessary, too. Some of the most important similarity measures are:

2.1.4.1.4.1 Measures derived from Minkowski's metric

$$d(x_i, x_j) = \left(\sum_{k=1}^{K} \left| x_{ik} - x_{jk} \right|^r \right)^{1/r} \quad r \geq 1$$

Where k is the number of input attributes. When r=1, *Manhattan* or *City-Block* distance function is obtained. If r=2, *Euclidean* distance is obtained. When including weights for all the attributes, the general formula becomes the following:

$$d(C_i, C_j) = \left(\frac{\sum_{k=1}^{K} weight_k^r * \left| d(A_{ik}, A_{jk}) \right|^r}{\sum_{k=1}^{K} weight_k^r} \right)^{1/r}$$

Where for not ordered attributes, their contribution to the distance is,

$$d(A_{ik}, A_{jk}) = 1 - \delta\big(qlv(A_{ik}), qlv(A_{jk})\big)$$

and δ is the δ of Kronecker.

2.1.4.1.4.2 Unweighted similarity measures

These two similarity measures ignore attribute's weight and are defined as follows:

Clark:

$$d(x_i, x_j) = \sum_{k=1}^{K} \frac{\left| x_{i,k} - x_{j,k} \right|^2}{\left| x_{i,k} + x_{j,k} \right|^2}$$

and *Canberra*:

$$d(x_i, x_j) = \sum_{k=1}^{K} \frac{|x_{i,k} - x_{j,k}|}{|x_{i,k} + x_{j,k}|}$$

2.1.4.1.4.3 Heterogeneous similarity measures

The following two distance measures show very high values of efficiency. These functions were proposed in [Wilson-Martínez, 1997] :

1. Heterogeneous Value Difference Metric (HVDM):

$$HVDM(i, j) = \sqrt{\sum_{a=1}^{m} d_a{}^2(x_a, y_a)}$$

Where *m* is the number of attributes. The function $d_a(x_a,y_a)$ returns a distance between the two values *x* and *y* for attribute *a,* and is defined as:

$$d_a^2(x_a, y_a) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown, otherwise} \\ normalized\_vdm_a(x, y), & \text{if } a \text{ is nominal} \\ normalized\_diff_a(x, y), & \text{if } a \text{ is linear} \end{cases}$$

Where $normalized\_vdm_a(x,y)$, is defined as follows:

$$normlized\_vdm(x, y) = \sqrt{\sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^2}$$

Where:

$N_{a,x}$ is the number of instances that have value *x* for attribute *a*;
$N_{a,x,c}$ is the number of instances that have value *x* for attribute *a* and output class *c*;
C is the number of output classes in the problem domain

The function $normalized\_diff_a(x,y)$, is defined as showed below:

$$normalized\_diff_a(x, y) = \frac{|x - y|}{4\sigma_a}$$

where σ_a is the standard deviation of the numeric values of attribute *a.*

2. Interpolated Value Difference Metric (IVDM):

$$IVDM(x, y) = \sum_{a=1}^{m} ivdm_a(x_a, y_a)^2$$

Where $ivdm_a$ is defined as:

$$ivdm_a(x,y) = \begin{cases} vdm_a(x,y) & \text{if } a \text{ is discrete} \\ \sum_{c=1}^{C} \left| p_{a,c}(x) - p_{a,c}(y) \right|^2 & \text{otherwise} \end{cases}$$

where $vdm_a(x,y)$ is defined as follows:

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| P_{a,x,c} - P_{a,y,c} \right|^2$$

C is the number of classes in the casebase. $P_{a,x,c}$ is the conditional probability that the output class is c given that attribute a has the value x. And:

$$P_{a,x,c} = \frac{N_{a,x,c}}{N_{a,x}}$$

Where $N_{a,x}$ is the number of instances that have value x for attribute a; $N_{a,x,c}$ is the number of instances that have value x for attribute a and output class c.

$P_{a,c}(x)$ is the interpolated probability value of a continuous value x for attribute a and class c, and is defined:

$$P_{a,c}(x) = P_{a,u,c} + \left(\frac{x - mid_{a,u}}{mid_{a,u+1} - mid_{a,u}} \right) * \left(P_{a,u+1,c} - P_{a,u,c} \right)$$

In this equation, $mid_{a,u}$ and $mid_{a,u+1}$ are midpoint of two consecutive discretized ranges such that $mid_{a,u} \leq x < mid_{a,u+1}$. $P_{a,u,c}$ is the probability value of the discretized range u, which is taken to be the probability value of the midpoint of range u. The value of u is found by first setting $u = discretize_a(x)$, and then subtracting 1 from u if $x < mid_{a,u}$. The value of $mid_{a,u}$ can be found as follows:

$$mid_{a,u} = min_a + width_a * (u+.5)$$

2.1.4.1.4.4 Weight-sensitive similarity measure: L'Eixample distance measure

An exponential weighting transformation would be required for a better attribute relevance characterisation, when the number of attributes is very high. L'Eixample [Sànchez-Marrè et al, 1998], [Núñez et al, 2002] is a normalised weight-sensitive distance function. It takes into account the different nature of the quantitative or qualitative values of the continuous attributes depending on its relevance.

L'Eixample distance is sensitive to weights. For the most important continuous attributes, that is weight > α, the distance is computed based on their qualitative values. This implies that relevant attributes having the same qualitative value are equals, and having different qualitative values are very different, even when a continuous measure would be very small. And for those less relevant ones, that is weight $\leq \alpha$, the distance is computed based on their quantitative values. This implies that non-relevant attributes having the same qualitative value are not equals, and having different qualitative values, are more similar. *L'Eixample distance* used to rank the best cases is:

$$d(C_i, C_j) = \frac{\sum_{k=1}^{n} e^{w_k} \times d(A_{ki}, A_{kj})}{\sum_{k=1}^{n} e^{w_k}}$$

where

$$d(A_{ki}, A_{kj}) = \begin{cases} \dfrac{|\, qtv(A_{ki}) - qtv(A_{kj}) \,|}{upperval\ (A_k) - lowerval\ (A_k)} & \text{if } A_k \text{ is a continuous attribute and } w_k \leq \alpha \\ \dfrac{|\, qlv(A_{ki}) - qlv(A_{kj}) \,|}{\#\,\mathrm{mod}(A_k) - 1} & \text{if } A_k \text{ is a continuous attribute and } w_k > \alpha \\ & \quad \text{or } A_i \text{ is an ordered discrete attribute} \\ 1 - \delta_{qualval(A_{ki}),qualval(A_{kj})} & \text{if } A_k \text{ is a non ordered discrete attribute} \end{cases}$$

and,

C_i is the case i; C_j is the case j; W_k is the weight of attribute k; A_{ki} is the value of the attribute k in the case i; A_{kj} is the value of the attribute k in the case j; qtv(A_{ki}) is the quantitative value of A_{ki}; qtv(A_{kj}) is the quantitative value of A_{kj}; A_k is the attribute k; upperval(A_k) is the upper quantitative value of A_k; lowerval(A_k) is the lower quantitative value of A_k; α is a cut point on the weight of the attributes; qlv(A_{ki}) is the qualitative value of A_{ki}; qlv(A_{kj}) is the qualitative value of A_{kj}; #mod(A_k) is the number of modalities (categories) of A_k; δqlv(A_{ki}),qlv(A_{kj}) is the δ of Kronecker.

2.1.4.2 Case adaptation

When the best partial-matching case selected from the Case Library does not match perfectly with the new case, the old solution needs to be adapted to fit more accurately the new case solution. This reusing process can happen during the solution formulation (adaptation), or after some feedback has pointed out some problem in the evaluation step, which needs to be fixed (repair).

There are many strategies that have been used in case-based reasoners. However, all these techniques can be grouped [Kolodner, 1993; Riesbeck and Schank, 1989] as 'null adaptation', 'structural adaptation' and 'derivational adaptation' - although in most case-based reasoners mixed adaptation methods are implemented.

Null adaptation is probably best suited in case-based systems with very simple actions in the solution (like accept/reject, a fault diagnosis, etc.) such as the first adaptation method used in the PLEXUS system [Alterman, 1988]. In those systems, the old solution is applied directly to the new case.

There are several *structural adaptation* methods, where the adaptation process is directly applied to the solution stored in a case. The structural adaptation methods can be divided into three major techniques: 'substitution methods', 'transformation methods' and 'special-purpose adaptation heuristics' or 'critic-based adaptation methods'.

<u>Substitution methods</u> provide the solution of the new case with appropriate components or values computed from components or values in the retrieved solution. Most outstanding substitution techniques are: parameter adjustment or parameterised solutions, where the differences between the values of the retrieved case and those ones of the new case are used to guide the modification of the solution parameters in the appropriate direction. This approach has been used, for example, in HYPO [Ashley, 1990] and PERSUADER [Sycara, 1987], JUDGE [Bain, 1986].

Other kinds of methods, such as direct reinstantiation used in CHEF [Hammond, 1989], local search used in JULIANA [Shinn, 1988], PLEXUS and SWALE [Kass and Leake, 1988], query memory used in CYRUS [Kolodner, 1985] and JULIANA, specialised search used in SWALE, etc., can be named as abstraction and re-specialisation methods. When there is a component like an object or a value from the retrieved solution, which does not fit in the new problem, these methods look for abstractions of that component of the solution in a certain knowledge structure (concept generalisation tree, etc.) that do not have the same difficulty; the last kind of substitution methods is the case-based substitution methods. They use the differences between the new and the retrieved case to search again cases from the Case Library to eliminate these differences. These techniques have been used, for instance, in systems such as CLAVIER [Hennessy and Hinkle, 1992], JULIA [Hinrichs, 1992], CELIA [Redmond, 1992], etc.

Transformation methods use either some common-sense transformation rules such as deleting a component, adding a component, adjusting values of a component, etc.), as in JULIA system, or some model-guided repair transformation techniques based on a causal knowledge, such as in KRITIK [Goel and Chandrasekaran, 1992] or CASEY [Koton, 1989] systems.

Special-purpose adaptation techniques or critic-based adaptation methods are based on some specific rules of repairing, called critics [Sacerdoti, 1977; Sussman, 1975], like those used in PERSUADER. Other systems such as CHEF and JULIA, use some domain specific adaptation heuristics and some structure modification heuristics.

Derivational adaptation methods do not operate on the original solutions, but on the method that was used to derive that solution. The goal is rerunning the same method applied to derive the old solution, to re-compute the solution for the new case. This methodology was first implemented in ARIES system, and was named as derivational replay [Carbonell, 1986]. In such a techniques, reinstantiation occurs when replacing a step in the derivation of the new solution, like in systems such as PRODIGY/ANALOGY [Veloso and Carbonell, 1993], JULIA [Hinrichs, 1992] or MEDIATOR [Kolodner and Simpson, 1989].

2.1.4.3 Case Evaluation

This step is one of the most important steps for a case-based reasoner. It gives the system a way to evaluate its decisions in the real world, allowing it to receive feedback that enables it to learn from success or failure.

Evaluation can be defined as the process of assessing the 'goodness' or performance of the proposed solution for the new case derived from the solution of the best similar remembered case. The evaluation process can point out the need for additional adaptation - usually called repair - of the proposed solution, although this only makes sense in non real-time domains. Commonly, this evaluation step can be performed either by asking a human expert (oracle) whether the solution is a good one or not, or by simulating the effects of the proposed solution in the real world such as in most planning or design domains, or by directly getting a feedback on the results of the proposed solution from the real world.

2.1.4.4 Case Learning

Learning is an interesting and essential cognitive task of case-based systems. There are two major kinds of learning in a case-based system:

- learning by observation;
- learning by own experience.

2.1.4.4.1 Learning by observation

Learning by observation happens when the system is provided with a set of initial cases, either by an expert or by direct observation (experience) of real data. Also, it can learn a new case by direct observation provided by an expert in any moment.

It is important to note that a case-based reasoner starts with a representative set of cases. They are like the training set of other supervised machine learning methods. To this end, the initial Case Library is usually 'seeded' with some situations obtained by classification procedures of historical databases. See for example [Sànchez-Marrè *et al.*, 1997b].

From these new discovered classes, some objects (cases) belonging to each class are selected to be included in the initial Case Library.

2.1.4.4.2 Learning by experience

Learning by (own) experience is done after each cycle of the case-based reasoner, offering the opportunity to increase the problem solving capabilities of the system - so, it can learn from the new experience. If the proposed solution has been a successful one, the system can learn from this fact, in the sense that if this experience is stored in memory, when a new similar case to this one appears, it can be solved as the past one (learning from success). If the system has failed, it must be able to prevent itself from making the same mistake in the future (learning from failure). It should be noted that not all the case-based systems have both kinds of learning.

2.1.4.4.3 Learning from success

When the new case has been successfully solved, the main option to follow is to store this new case in the Case Library. That is, inserting the new experience in the appropriate place into the Case Library so that it can be recalled when it can be more useful, and cannot be recalled indiscriminately. In other words, the case must be placed in the neighbourhood space of the memory where it would be easily recalled in the retrieval step. Thus, good indexes must be chosen to implement this strategy. While the new case is being placed in the Case Library, memory's indexing structure and organisation is updated appropriately.

2.1.4.4.4 Learning from failure

Assuming that the adaptation method is correct, two reasons could originate a failure. One is that the case retrieved is the best one for solving this new situation, given the current Case Library, although it is not very similar. The problem here is that there are not enough cases (experiences) in the Case Library to cover the whole space of cases. The solution relies on learning a lot of new relevant experiences to store in the Case Library. In hierarchical case libraries it can happen that although a very similar case to the new one exists, it has not been retrieved. Thus, there is something wrong in the retrieval process. Perhaps the similarity assessment is not well suited. Perhaps what is wrong is the discrimination tree/network organisation. If the discriminating power of attributes is not good, then the retrieval algorithm can miss the best similar case due to the fact that it will be searching in some other region of the tree, where the best similar case is not there. The solution to this problem relies in re-organising the Case Library's structure [Veloso & Carbonell, 1993].

When the new case has failed, there are several possible actions to be taken in order to ensure that this failure cannot be repeated in future. First, the case-based system can store the failed case in its memory in such as way as to prevent taking the same failed solution for similar cases at another time. Some case-based systems maintain a separate Case Library of failed cases [Hammond, 1989], and others maintain only one Case Library structure.

In a failed case library system, a previous step is added to the general CBR cycle, known as the 'anticipation phase'. Before retrieving any successful case from memory, the failed case library is first checked for matches to the input new case to avoid repeating the failure. In the single case library structure, the equivalent of that anticipation step is implemented as a 'filtering task' applied to the searched cases in the retrieving process. Previous failed cases are eliminated and the cases that were the source experience to derive those failed cases removed from the list of retrieved cases. Thus, the system can avoid making the same incorrect action that it did in past situations.

An additional action could be to incorporate to the system's memory the right solution as defined by a human expert, enabling this recorded experience to be remembered and used appropriately in the future.

Other tasks that can be performed are updating the weight of the attributes and thus modifying the similarity measure such as in [Bareiss, 1989; Koton, 1989], or changing their order in the discriminating list. Another feature is to update the utility measure of the retrieved cases that could derive the new case.

2.1.5 Positioning CBR

The aim of this section is to compare and contrast CBR with techniques derived from information retrieval, statistics, pattern recognition, knowledge-based systems, machine learning and neural networks.

2.1.5.1 Positioning CBR and Statistics

2.1.5.1.1 CBR and Linear Discriminant Analysis

A comparative study between linear discriminant analysis and CBR was conducted at Daimler-Benz with respect to a gearbox quality control problem [Hübert & Nakhaeizadeih, 1993]. There were 7,080 cases in the case base, described by 56 attributes. These cases were partitioned into 91 different classes (diagnoses). The results were evaluated through a cross-validation procedure where 80% of the case base was used for learning and the remaining fifth for testing. By selecting different test and learning sets randomly each time, Daimler-Benz calculated the average of the results for five different tests. The following table summarises the results in terms of the average percent of good results.

Tests	Case-Based Learning Algorithm	Linear Discriminant Analysis
1	93.4	61
2	93.4	60
3	92.6	62
4	93.2	61
5	93.5	62
Average	**93.2 %**	**61 %**

Table . Comparison between Case-Based Learning and Linear Discriminant Analysis

The statistical method performed poorly compared to the case-based learning method. Linear discriminant analysis requires a sizeable amount of data for correct operation (for example, to estimate the median of each class and the global covariance matrix). Theoretically, for 56 attributes and 91 classes 1,603 parameters have to be computed. Even with a large number of cases, all these parameters cannot independently be estimated. Hence, the linear discriminant function is based towards the first-estimated parameters.

2.1.5.1.2 Comparison between CBR and Statistics

In the previous sub-section we presented applications where CBR and induction-based techniques performed significantly better than statistical methods. However, it would be misleading to state that CBR is always better than statistics. Statistics and CBR are complementary techniques in many problem-solving processes. Statistics work well on large amounts of standardised data to test known hypotheses. However, most statistical methods are not suited for exploratory analysis (i.e. when all hypotheses are not yet known) because they require strong underlying critical assumptions that are often overlooked by the end-user (for example, the independence of attributes).

In addition, when using statistical methods it is hard to take into account common-sense or background knowledge. CBR on the other hand can make use of background knowledge when available since it integrates numeric as well as symbolic techniques.

2.1.5.2 Comparison with Information Retrieval

Although CBR and Information Retrieval (IR) have a great deal in common and are often used for similar tasks, there are relatively few studies comparing the two techniques. This may be due to the fact that CBR and IR originated from, and were developed in, two different communities.

CBR like IR focuses on retrieving relevant information from a database (case base) of collected data. Both allow flexible databases querying and result in a collection of relevant by inexact matches. This is the main difference between the two technologies and relational database systems. CBR and IR differ in the following way:

- Data type - Whereas IR methods mainly operate on textual data, traditional CBR methods operate on vectors of several basic data types: real, integer, symbol, Boolean, string, etc.
- Amount of data - IR methods can handle huge amounts of data. IR can search thousands of documents, consuming gigabytes of memory. CBR systems are comparatively more limited.
- Use of knowledge - IR systems operates without knowledge of the user's problem-solving task. They provide a generic indexing and retrieval engine that can be used for a wide range of tasks. As a consequence of this they have limited accuracy for any given query. CBR systems, on the other hand, make use of knowledge about the problem-solving process in order to build effective indexes, such as decision trees or k-d trees and to improve retrieval accuracy.

These difference are true in a shallow comparison of the CBR and IR systems. However, in a deeper comparison, the differences become blurred. Current IR tools often operate on mixed data types and use a thesaurus or concept hierarchies during retrieval. On the other hand, some commercial CBR tools do not represent and use background knowledge. They often only use a similarity function on flat attribute value vectors.

Therefore, it is better to say that CBR tools are primarily concerned with mixed representations whereas IR systems are primarily concerned with textual databases. Furthermore, CBR tools often explicitly represent the knowledge they use, whereas IR systems do not. Hence, it is possible to consider intensive problem solving and learning methods (e.g. in synthetic applications domains), where the difference between CBR and information retrieval becomes very apparent. However, for application tasks such as decision support, help-desk systems and diagnosis, where syntactic approaches to similarity assessment and simple reuse strategies are often sufficient, the differences between a knowledge-based IR system and a low-level CBR tool are minor, especially when compared to knowledge-based approaches to information retrieval.

2.1.5.3 Comparison with Rule-based Expert systems

Developing rule-based expert systems that can solve complex real world problems is a difficult task. One of the main difficulties is due to the fact that rules have to be provided by human experts. Human experts are very good at solving practical problems, but not so gifted at explaining how they have solved a particular problem. In addition, they can seldom articulate this knowledge using logical rules that can be expressed in a formal language.

CBR provides methodologies for building, validating and maintaining applications. Instead of providing rules, specialists talk about their domain by giving examples; it is more intuitive to answer a question such as “Have I ever seen this problem before?” than to provide a general definition of a class of problems. Rules handle big chunks of the problem domain well but perform poorly on boundary regions where experience needs to be accumulated on a daily basis. CBR is valuable when problems are not fully understood (weak models with little background knowledge available) and where there are many exceptions to the rules. In such situations the number of special or subtle contexts makes a rule-based approach inadequate. A typical example of a purely case-based approach in a weak domain is the CLAVIER application.

Finally, methods based on cases are incremental. They can learn from experience and keep up with the knowledge that workers acquire in their daily experience. This maintenance task is much more difficult when using a rule-based system: as the system expands, smaller and smaller chunks of the domain have to be incrementally covered by rules. This results in declining productivity, increasing difficulty in maintaining the rule base and, ultimately, leads to an incomplete coverage of the problem.

The following table gives a rough classification of methods that use cases and/or rules.

	Exact Matching	Partial Matching
RULES	Standard Rule-Based Approach	Analogy-Based Approach
CASES	Standard Database Approach	Case-Based Approach

According to the complexity and/or the goal of the task to be achieved, it is possible to evolve smoothly from a rule-based system - where all the possibilities of decision making are sketched, to a purely incremental case-based system - where cases model the entire problem domain. A helpful and promising compromise can be found in hybrid systems where the domain is modelled with rules as far as possible and boundary regions are handled by cases.

2.1.5.4 Comparison with Classical Machine Learning Approach

There is no general consensus about the overall learning task that is addressed by CBR. A major distinction is that machine learning systems make a strong separation between learning and problem solving. Learning involves analysing training examples to extract functions or rules; problem solving involves applying these functions to new incoming problems. In contrast, CBR does not separate the two. However, the transformation of a simple learning algorithm into an equivalent case-based variant underlines in principle the equivalence of symbolic and case-based methods. The equality of the learning power of symbolic and case-based classifiers is even proven for the area of inductive inference. The proof is based on the learner's ability to adjust the similarity measure to the given problem. There is no theoretical difference between case-based classifications compared to the traditional symbolic learning approach in this simple framework. Both mechanisms can learn the same concepts. Nevertheless, work in CBR is concerned with complete systems, whereas work in symbolic machine learning is more concerned with algorithms. In addition, CBR explicitly includes the notion of memory that eases the mapping onto practical problems.

In practice, an important difference between case-based and symbolic classification algorithms concerns the representation of the learned concept. The symbolic approach corresponds to a kind of compilation process whereas the case-based approach may be viewed as a kind of interpretation during runtime. Which approach should be used in any given situations depends mainly on the simplicity and adequate of the representation of the given knowledge to the application to be delivered.

2.1.5.5 Comparison with Neural Networks

Neural networks perform better than CBR in a knowledge-poor environment when the data cannot be represented symbolically - as in RADAR signal recognition. The field of application of Neural nets also extends to pattern recognition where there are many points of raw data, as in vision, speech and image processing. Neural networks are very resilient to noise during the consultation phase. For instance, even with only a fractions of the original attributes having values, retrieval performance can still be very high. However, Neural nets have drawbacks that have yet to be removed. Firstly, they are not suitable when background domain knowledge has to be taken into account. Secondly, they cannot cope with complex structured data. Furthermore, in order to perform well - efficient training and good convergence - the coverage of the domain has to be exhaustive during the "learning" phase.

Ergonomically, Neural networks suffer from a lack of transparency. Users cannot judge the validity of the network's decisions because of the nature of its inner workings. This is because the output of the network is a function of weighted vectors that depends on the network's architecture and the learning mode used. No justification or explanation of its output can be easily provided. On the one hand, the classic bottleneck of the problem of questioning field specialists is avoided (as is the case with other machine learning paradigms), however, such experts can neither validate nor modify the resulting system.

Neural networks and CBR can be integrated within a larger system. For instance, in the PATDEX system [Althoff & Wess, 1991], the relevance of the attributes with respect to a given diagnosis is updated through a competitive learning mechanism derived from unsupervised Neural networks. The technologies also complement each other in that each is suited to different types of application. Depending on the amount of available knowledge and on the goal of the target application, the appropriate technology can range from a low-level knowledge approach where little explanation is required (neural nets), to a rich and complex learning scheme that involves high-level learning techniques.

2.2 MBR BACKGROUND

2.2.1 Introduction

In general, symbolic reasoning methods are often divided into three groups (e.g. [Luger-02]): model-based, rule-based, and case-based reasoning. The two first are based on reasoning from general domain knowledge, while the latter is based on situation-specific knowledge. The Model-based Reasoning process itself can be viewed as the symbolic processing of an explicit representation of the internal workings of a system in order to predict, simulate, and/or explain the resultant behaviour of the system from the structure, causality, functional and behaviour of its components. Just as rule-based methods, model-based methods may or may not allow for incompleteness and uncertainty.

As a general rule, Model-Based Reasoning (MBR) concentrates on reasoning about a system's behaviour from an explicit model of a mechanism by modelling that behaviour. Because they employ models that are compact axiomatic systems from which large amounts of information can be deduced, MBR techniques can very succinctly represent knowledge more completely and at a greater level of detail than techniques that encode experience.

Model-based reasoning does not restrict to AI fields as it is, in fact, a technique widely used by engineer and scientist of all fields, as well as economist and politics, for example. It can be based on simulation techniques, but it can as well not even involve computers. Analytical modelling for example is quite broadly used for simple systems. And in the aeronautical field, material objects, reduced copies of the original, are often used to reason and perform experiments that help analyse the original system.

Like CBR and other AI techniques, human is the first “model” of inspiration. MBR could be one of the basic mechanisms used by humans for knowledge acquisition. Human learn by progressively refining their model of reality. They make assumptions, imagine and memorise a first approximated model, try it out by conducting experiments on real cases, and revise their internal model. This process can be viewed as a mixture of CBR and MBR. MBR is about using acquired concepts to analyse reality while CBR is about acquiring aspects of both techniques clearly, naturally shows-up.

The model-based approach to diagnosis emerged in the early 1980's as a solution to certain limitations in the traditional expert system approach based on heuristic knowledge. Since then, research in knowledge-based systems has very much focused on exploiting model based reasoning methods in order to overcome limitations due to previously used expert systems technology based on empirical associations. The core of the current technology is an explicit, declarative model of a technical system that can be composed from context independent re-usable behaviour models of components taken from a domain library. Methods for formulating and using such models at a conceptual and qualitative level have been developed in the area of qualitative reasoning [Weld-de Kleer, 1990], [Faltings-Struss, 1992], [Dague et. al. 1990], [Lee-Ormsby, 1994], [Struss 1988, 1989, 1992]. Automated diagnosis is the task for which the most advanced MBR work has been carried out [Console-Hamscher-de Kleer, 1992].

2.2.2 Origins of MBR

We can find in the history of MBR techniques two different origins, and as a result of mixing both, a third one that we called *Hybrid*:

2.2.2.1 <u>Consistency-Based systems</u>

Some MBR are uniquely based on a correct model of the system and rely on sophisticated reasoning method to efficiently find out what is going wrong in the system with no prior knowledge at all on the faulty behaviour of the system. The diagnostic is possible whatever the defect is and often the actual internal faulty behaviour of the system is retrieved, providing a strongly supporting explanation of the diagnostic (for a digital circuit, if it is found out that a component does behave as a multiplier whereas it should have been an adder, it becomes clear that somewhat mistakenly put a multiplier instead of an adder when building the digital circuit – a fault tree model will never solve out such problems). Such MBR are said **consistency-based**. As we shall see, it is a term with a precise meaning that belongs to the logic vocabulary. Two simultaneous works founded consistency-based MBR:

- [Kleer & Williams, 1992] research, based on a application of troubleshooting electronic circuits, appears to be among the earliest approaches in the literature. Their system, General Diagnostic Engine (GDE) became a standard and is still a reference for most currently existing systems. Their work is based on "sound intuition" rather than formalism and aim at proving the concept by presenting a fully functional application.
- In the other hand, Reiter's work [Reiter 1992] [Greiner, Smith et al. 1992] aimed at providing a formal theory that support the method and does not describe a functional system.

2.2.2.2 <u>Abductive systems</u>

Others MBR are an evolution of fault tree systems. In such systems, besides the correct model, the model is also told what are the probabilities of failure and the corresponding expected behaviours of the system components. To each component corresponds a behaviour mode, including a correct one, with a given probability of occurrence. When a discrepancy is found, it must be explained in term of a list of components working in a specific faulty mode. Such model is said of abductive type, a term that belongs again to logic, because you have to retrieve causes given a list of effects. **Abductive systems** are more restrictive than consistency-based systems since in case of an unplanned failure, they will probably fail to find-out any cause or, give a diagnostic too imprecise (much more components assumed to be faulty, and to be checked-out, than necessary) or even give wrong answers. Nevertheless, abductive systems are faster than consistency-based systems.

2.2.2.3 <u>Hybrid systems</u>

Not surprisingly, the tendency is to try to get the best of both techniques. Therefore, systems are rarely pure and hybrid systems can be constructed from extensions of the systems described above:

1. Extensions of consistency-based diagnosis deal with fault model. For instance, components are assigned several modes: one normal mode, one or more faulty ones and often, one unknown mode. Unknow modes are keeping the method in its original spirit. If the confidence in the fault model is important, probabilities for unknown modes can be assigned very low values.

2. Extensions of the abductive diagnosis deal with a correct model of the device, besides the fault model.

2.2.3 Models

2.2.3.1 <u>Some previous definition</u>

In order to clarify concepts that will appear in the proper explanation of the model and modelling techniques, the following definitions are posed:

2.2.3.1.1 What is a system?

Among others, we found:

1. A regularly interacting or interdependent group of items forming a unified whole
2. A group of devices or artificial objects or an organization forming a network especially for distributing something or serving a common purpose
3. An organized set of doctrines, ideas, or principles usually intended to explain the arrangement or working of a systematic whole <the Newtonian *system* of mechanics>
4. An organized or established procedure <the touch *system* of typing>
5. A manner of classifying, symbolizing, or schematizing <a taxonomic *syste*m> <the decimal *syste*m>

[Cellier, 1991] prefers to underline only some basic properties of systems. We retained:

6. The largest possible system of all is the universe.
7. A new system is apiece of the universe. It is characterized by the fact that we can say what belongs to it and what does not, and by the fact that we can specify how it interacts with its environment (inputs and outputs).
8. System definition can furthermore be **hierarchica**l. We can **cut it out in smaller par**t, an we have some new systems.
9. It is also hierarchical in a sense that it involves making some simplifications : often if not always, it is impossible (or unnecessary or inappropriate) to capture all details of a given system prior reasoning about it. Several representation (we will say son models) of a given system, each with a different level of details can be used.
10. A system interact with its environment: it can be controlled and observed.

[Cellier, 1991] also cite a very concise definition of a system: “A system is a potential source of data”.

2.2.3.1.2 What is an experiment?

1. “An experiment is the process of extracting data from a system by exerting it through its inputs”
2. It consists in making use of its property of being “controllable” and “observable”
3. That is to apply a set of external conditions to the accessible inputs and to observe the reaction of the system to these inputs by recording the trajectory behaviour of the accessible outputs.
   - A difficulty of system experimentation is that real system are usually under the influence of a large number of additional inaccessible inputs (so-called disturbances), and that a number of useful outputs are not accessible through measurements either (they are internal state of the system).
   - One of the major interest for simulation is the fact, that, in the simulation world, all inputs and outputs are accessible. It is also possible to execute simulation outside the range of experiments that are applicable to the real system.

2.2.3.1.3 Some other definitions

In order to clarify the concept of a model it is useful to distinguish it from other closely related concepts:

- **Analogies:** An analogy or metaphor is established simply by creating a set of points of similarity or correspondence. Analogies are limited and break down if extended.
- **Data laws:** These refer to the law-like properties of experiential data that are often perceived as invariance or 'patterns' in the data. This is a convenient way to represent the data with reduced detail but the pattern selected is one off many possibilities and does not by itself imply that the law used has any justification as a mechanism or theory underlying the data.
- **Hypotheses:** A hypothetical model is one that offers a view of the object or system that may be helpful in structuring or understanding the underlying phenomena. A hypothesis offers a provisional explanation of a system and its behaviour but, being tentative, can only weakly claim to represent a theory of the real mechanisms operating in the modelled world.
- **Conjectures:** These are forms of representation that make claims to describe some true state of affairs and can act as a theory until their context becomes inappropriate or they are refuted. They may be seen as theories with weak evidence. Several conjectures may compete.
- **Theories:** Theories are the most committed representation as they claim to represent the actual governing principles of the object and are required to be taken as correct and true. We define a theory as a systematic set of propositions, principles and rules about a section of some domain that captures its central, distinctive features and has the potential to increase understanding through some form of interpretation. Theories are assumed correct until disproved by new knowledge or replaced by better theories. This means alternatives cannot coexist and it is not sufficient for a theory to be either utilitarian or approximate.
- **Models:** A first overview of what a model is, regarding the definitions above, shows that a model contains or embodies a selected set off assumptions about the modelled object and can thus exhibit particular properties of the object. They are distinguished from analogies by their need to satisfy a particular purpose. Models normally have some grounding in a domain theory and are able to generate predictions or display consequences of the theory.

2.2.3.2 <u>What is a model?</u>

One of the central questions is what precisely counts as a model. According to the *Platonist* conception, a model is a type that constitutes the best, ideal, standard or paradigmatic specimen of a certain class, which may be implemented in more than one token, e.g. a model husband. As such, it can work as a criterion for evaluation and may have an axiological value. In mathematical logic, where a theory T is understood as a set of well-formed formulae according to a language L, a model is an implementation of T in a language different from L, usually set theory, in which every member of T is satisfied. Following Giere [Giere, 1998], this can be called the *instantial* conception of a model. It is the one proposed by Patrick Suppes in his seminal work on the philosophy of models. Both the *Platonist* and the *instantial* conceptions are *essentialist*: they consider models special "things", theoretical entities. Arguably, they are both less interesting than the *functionalist* conception, according to which a model is actually a synecdoche, used to refer to the more important relation of *modelling*.

Modelling (**M**) is a ternary relation of correspondence between a (modelled) system **s**, a (modelling) structure **m** (the model strictly speaking; note that systems are usually unstructured and this is why models are also known as structures) and an interpretation **i**, that maps the most significant information elements in **s**, both static and dynamic (typically, classes of objects, properties, functions, relations, actions and processes), onto information elements in **m**, that is **M** (**s**, **m**, **i**). The relation of "correspondence" is to be preferred to the more standard "representation" because only the former leaves open both the alethic and the functionalist approach. If **m** counts as a reliable model of **s**, then in the former case **m** describes or mirrors at least some essential features of **s** in a sufficiently accurate way, like a map representing a territory. In the latter case, not only can **m** represent **s**, but it can also fit or grasp at least some essential features of **s** in a sufficiently accurate way. In this case, the relation resembles the way in which a key fits a lock: the pre-modelled system works as a receptor and the model is "keyed to" (i.e. brought into harmony or conformity with, made appropriate or attuned to) the system, through an appropriate interpretation. This is not to be understood in terms of an instrumentalist epistemology, but rather as an argument for indirect realism: some of the properties of the receptor (the lock) may be indirectly understood by analysing the nature of the model (the key).

When **M** is satisfied, **m** is described as a model of **s** under the given interpretation **i**. Sometimes, the interpretation **i** is embedded in **m**, as it is the case with representational models such as the map of the London Underground. Sometimes it is external: in this case a typical example is a collection of billiard balls (**m**) in random motion, explicitly interpreted (**i**) as an analogue model of an ideal gas (**s**). Note that a collection of billiard balls or a network of lines and nodes *per se* are not models of anything.

Critical constructionism in epistemology and, more specifically, the semantic approach in the philosophy of science adopt a *functionalist* (Giere's *representational*) analysis and argue that knowledge, including scientific theories, consists of representing structures that model systems. A stronger claim may follow: the history of knowledge can be interpreted as the diachronic evolution of modelling structures.

Reading through the various essays in the book, it becomes clear that four main questions confront the *constructionist* approach:

1. **the abstraction problem**: is there any method to individuate the salient information elements in **s** that need to be mapped onto **m** to ensure that **m** is actually an adequate model of **s**?
2. **the heuristic problem**: how can the simplification of **s** and the **s**-independent information elements (if there are any) in **m** contribute to the advancement of scientific research?

A third question can be clarified by adopting Peirce's terminology. **M** is an *iconic* not a *symbolic* relation. A modelling structure refers to a modelled system more in the way that the image of a skull is used to indicate some danger, than like an arbitrary and conventional sign, such as a coloured flag, which can refer to a certain kind of weather only conventionally. So the third question becomes

3. **the hermeneutic problem**: what kind of representational relation(s) (similarity, resemblance, analogy, isomorphism…) coordinate(s) **m** with **s**?

A slight revision of (3) leads to the debate on realism and anti-realism in the philosophy of science:

4. **the truth problem**: is there a precise extent to which an epistemically adequate and instrumentally successful scientific theory, understood as a model, may be said to correspond to the actual nature of the real system (i.e. the designated part of reality) it refers to?

As regarding (4), two comments are in order.

It is important to stress that the distinction between modelled systems **s** and modelling structures **m** is, to some degree, a question of methodological convenience. On the one hand, models can be treated as systems, not only in the standard mathematical sense (model theory), but also in the twofold sense that, metalogically, scientific work can be a matter of high-level modelling of models, and, technologically, models can actually be implemented. On the other hand, systems should not be naively viewed as uninterpreted "things in themselves", waiting to be properly represented, like unsemanticised empirical observations, i.e. raw data not yet interpreted as meaningful information. As error analysis clearly shows, for example, no theory deals with pure uninterpreted data; the reference of a model, even of a data model, is always information (meaningfully interpreted data), i.e. another model within an interpretative context. The difference between **s** and **m**, if properly understood, is therefore one of semantic and epistemic levels of conceptualisation and hence freedom of design, control and management (epistemic hierarchy), not ontic, as if **s** and **m** were two utterly different sorts of things, one "real" and the other "conceptual". The more a model is inextricably coupled with the reality it corresponds to, the more its specific conceptualisation becomes inevitable: it may require an explanatory model, but there is no level at which scientific or ordinary experience can have access to model-free data sources of information. It is in this sense that one may wholeheartedly agree with Giere when he writes that "reasoning about the world is primarily reasoning with models. […] It is models almost all the way up and models almost all the way down". Reality as we know it is the threshold where data endlessly collapse into information. The noumenal world remains far away, as a sort of apeiron, but this is not a problem, because science is interested in the construct, not in the indefinite source of possible experiences.

Concerning the relation between systems and models, a model is a key to a system in a twofold sense: it gives an explanation, interpretation or solution of the system, and by its means one gains entrance, possession, or control of the system. As such, it is never unconstrained and, although this feature is not prominent in the volume, "constraint" is a fundamental concept in the realism debate on models.

A primitive constraint is a boundary condition that restricts the range of applicability or interpretation of some statement. It is often written as a logical relation between $n \geq 2$ terms - typically a functional equation or inequality, each taking a value in a given domain; e.g. the whole equation [x = y + 1] is a constraint on x - but terms themselves (constants, variables or formulae with the form $f(x_1, ..., x_n)$, where f is a constructor of arity n, and $x_1, ..., x_n$ are terms) can also act as constraints. Valuation or constraint satisfaction is the process of assigning values to variables so that all constraints are true in a model. Primitive constraints can be combined into complex constraints (e.g. [[x = y + 1] and [$x \in$ N]]) and develop dynamically, interacting with the model (in Constraint Logic Programming, a very powerful modelling language, this is called "generalized constraint propagation"). A constraint set is a group of boundary conditions that have been applied to a model. Analyses may be run with one or more of these sets defined. Each represents a different model configuration and therefore will return different results. Now, hard constraints express necessary conditions and hence are negative requirements that an adequate model should not violate: they limit the search space for acceptable models of the analysed system by eliminating some alternatives from consideration. Soft constraints, on the other hand, are positive preferences, regulative ideals in Kantian terminology. They provide direction as to the optimal design of the model, so they are typically expressed in terms of maximisation or minimisation of some parameters established a priori, and can be satisfied to various degrees. If a model has n hard constrains, n is the degree of hard constrainedness. Strictly speaking hard constraints are Boolean, i.e. they can either be satisfied or not, only soft constraints are fuzzy, i.e. partly satisfiable, each soft constraint being associated to an element from a set, to be interpreted as a resource cost, or a level of preference, or a probability, etc. The absence of n hard constraints is indicated as the presence of an equivalent degree of freedom. Conversely, a hindrance to a degree of freedom is equivalent to a degree of hard constrainedness. Consistency is a classic hard constraint on a scientific theory.

2.2.3.3 Characteristics of a model

There are three key model characteristics:

- **Representation -** captures all the significant aspects of importance in the problem or application.
- **Prediction - t**he process of estimating how a system will perform in a given situation.
- **Explanation** - to increase confidence in predicted results it is necessary to justify them. An application may require that model behaviours be explained through an exposition of the mechanisms operating in the model. This may involve details of internal causality within the model.

Predictive power is necessary if a model is to be used for analysis and evaluation of the system under study. Prediction is usually achieved by deductive inference in some form of analysis process, often based on either mathematical or simulation techniques. But predictions are of little value if they cannot be explained or justified. This means there must be some interpretation.

Models must therefore contain sufficient 'causal' information to support the level of explanation desired which implies very rich models that are very expensive to create and to maintain - and this information richness implies more complexity.

By definition, all models, and indeed all forms of representation, must be less rich in knowledge or information terms than the objects represented. They are therefore necessarily approximations of the objects in some way. The main purpose of modelling is to take advantage of this reduction in data or detail in order to handle the great complexity found in many applications. There are three main ways that models reduce information content:

- in space - where removing small-scale detail, unnecessary features or irrelevant structure can reduce spatial data.
- in time - where using sampled time points, compressed state descriptions or less accurate variables can reduce temporal data.
- in complexity - where complexity is removed by eliminating variables or elements that do not contribute to the aspects of behaviour to be modelled.

However, by reducing conformation it is possible to loose accuracy and expressivity.

2.2.3.4 Modelling what?

An *object* is a distinguishable part of the world subject to our modelling task. This part of the world is called the *domain*. Objects can be material or abstract. At the lowest epistemological level a system is defined on the object by selecting relevant *attribute*s. They are chosen by the investigator and in accordance with the goals he/she intends to achieve, available resources, domain knowledge, the expected interaction with the rest of the world. The object is unique. We can define diverse attributes and investigate the same object from different viewpoints, i.e. we can define different systems on the same object.

In order to differentiate among individual observations of the same attribute we need some distinguishing property. As soon as we can index the observations it is possible to observe changes, tendencies, constraints or even general laws holding among attributes. In [Klir, 1985] the distinguishing properties are called *backdrop*s. They are usually time, space, population or their combinations. Each attribute and backdrop has associated a given set of values. For a given object the *object system* is defined as a set of attributes and a set of backdrops.

For example, let us suppose that the task is the technical diagnosis of a car engine, more specifically the carburetter adjustment. The *object* is a particular car. The task determines the selection of relevant *attribute*s. They are idle revs, emissions, fuel consumption etc. The *backdrop* here is time, since we are interested in these attributes before and after tune up. In this case the numerical value of time is not important, because we only need to distinguish an order of two measurements of attributes. Examples of other attributes and backdrops are: a humidity in various places of a geographical area (space) or the variation of parameters within a set of electronic components of a given type (population). The fact that time is the backdrop does not mean automatically that we are dealing with dynamic systems in the usual meaning (see the subsection below).

The attributes are mapped by means of *observation channels* into corresponding *specific variable*s, the backdrops are mapped into *specific support*s. The mapping must be homomorphic with respect to the observed properties of attributes and backdrops, i.e. it must preserve these properties. A *specific image system* is defined as the set of specific variables and the set of specific supports. The set of values on an attribute is mapped to a *state set* of the corresponding variable, the values of backdrops are mapped to a *support set* .

The observation channel can be either *crisp* or *fuzz*y. The crisp channel assigns a single value to the variable. The fuzzy channel assigns to each attribute a tuple of values with associated numbers from the interval <0, 1>. We call these numbers *degrees of certainty.*

2.2.3.4.1 Uncertainty

In general the topic of uncertainty has been intensely studied and lots of systems have been developed with higher or lower degree of success and accuracy. In this section we highlight just some of the most important and relevant to our focus of study.

Uncertainty and incompleteness in a model-based system may be captured as a numerical strength (“degrees of belief”) attached to a relationship, as for example in the Heart Failure Model from MIT [Long, 1996], and in Bayesian networks [Pearl J, 1990], [Pearl J, 2000]. Interesting work that combines the latter with a richer concept representation formalism is a current topic at Stanford University [Stanford-URL].

The work by [Heckerman et al., 1994] describes a system to generate troubleshooting plans in the face of uncertainty in the relationships among components and device status, observations, as well as the affect of actions on device status. A plan is considered as a set of observations and component-repair actions.

The Spook system [Pfeffer et al, 1999] reasons about the locations and status of military units based on intelligence reports. The domain modelled deals specifically with missile battalions, the batteries within those battalions, and the individual units (vehicles, radar emplacements, missile launchers, etc...) within the batteries. This project models a complex scenario dealing with uncertainty and imprecision.

2.2.3.4.2 Time

In principle, [Console & Torasso, 1992] is limited to static systems. It does account for causal relationships of the form, 10 minutes after component *C* is broken, the temperature start rising sharply. [Brusoni, Console et al. 1996] presents an extension of [Console & Torasso, 1992] spectrum, that would still hold in case of dynamic or more precisely temporal behaviour.

In fact he defines three possibilities:

- Time-varying context: The behaviour of the system is observed in different contexts, that is different times: for instance different inputs are provided to the system. The system could be assumed to maintain its faulty behaviour over time or could be allowed to change behaviour. Usually time-varying context is associated with one of the two other possibilities (temporal or time-varying behaviour).
- Temporal behaviour: the consequence of the fact that the system is in a specific (normal or faulty) mode manifest themselves after some time and for some time (during a period of time).
- Time-varying-behaviour :a system may have different faults across time.

Whereas the [Console & Torasso, 1992] system has only one dimension of choice:

- The logical explanation type (consistency-based versus abductive)

[Brusoni, Console et al. 1996] framework introduces another one:

- the notion of temporal explanation adopted (again: consistency-based versus abductive)

They also pay attention to stay very general as regards to the modelling language. They in fact propose a language (LATER) that can capture both temporal behaviour (including dynamic behaviour) and time varying behaviour and that is completely independent from the model of time adopted.

The overall idea to deal with time is to define time interval of validity for components behaviour, constraints and algorithms are based on Allen’s interval algebra [Allen, 1983].

“The completeness of the model is the criterion to choose abductive over consistency-based diagnosis. In particular, since abduction can be regarded as deduction on a completed theory, its use is recommended when the model of the system to be diagnosed is “complete” (i.e., when all the causes of the observations are included in the model). [...] Similar considerations apply also for the temporal dimension”.

Another approach is the work done in [J.M Gimeno et al, 1997] where they depicted a transition network (or automaton) for describing the temporal relationships of the different states of the process to the experts.

2.2.3.5 <u>Modelling how?</u>

For modelling the internal workings of a system or a specific domain, it is required to have a representation technique, expressive enough in order to achieve the reasoning objectives (simulation, explanation and/or prediction). Each concrete case has its own expressivity and accuracy requirements, and depending on them, one or another technique is used.

Automated learning algorithms exist supporting many different types of model classes: neural networks (eg. MLP, RBF, SOM), decision trees, association rules, episode rules, logic programs, Bayesian belief networks and Fuzzy inference systems (FIS). Learning may be divided into structural learning (system identification in classical system theory) and parameter learning (parameter estimation in classical system theory). Structure determines the flexibility of the model in the approximation of the mappings. There is a trade-off with good generalization ability and model complexity. More complex models may be fitted well to the training data but they tend to perform much worse on unseen validation data indicating low generalization capability. One should aim at building models with a suitable complexity.

There are different types of learning schemes available depending on the type of data that is available. In supervised learning, there exists a clear output (possibly provided by a reliable teacher) for each input data vector and the task is to learn an adequate input-out-put mapping from the data. Reinforcement learning may be applied in cases where the feedback is given occasionally, may be delayed and is only partially targeted. Unsuper-vised learning is used when there is no feedback available and the aim is to find internal structure within the input space. Typically clustering is used to form clusters of measure-ments where the similarity between vectors within a cluster is as close as possible and vectors from different clusters are as dissimilar as possible.

Typically the current automated learning algorithms require that all the knowledge from the system has been represented as input feature vectors with the same amount of features (all data needs to be on the same vector format). Possible missing values need to be pre-processed appropriately and the studied phenomena has to be represented in the vector format by calculating appropriate features describing the raw data.

Learning from data presumes the existence of measurement data records that represent the modelled phenomena adequately. There has to be enough data describing the system while it is at a stable state. The measurements should describe the same phenomena. However, some kind of models can be constructed directly from the knowledge of the experts.

Each day new techniques for modelling appear. These techniques are, in mostly all cases, ad-hoc designed for being optimal in a concrete problem. However they can be clustered in more general families of techniques. In this subsection we present some of the families of general modelling techniques that we consider useful for the achievement of our targets within RIMSAT context.

2.2.3.5.1 Transition networks (Causal-Temporal)

Transition network is a concrete technique for capturing in an automaton, causal and temporal knowledge. They were used in the environmental domain by [Gimeno et al, 1997]. They proposed a methodology to induce automata descriptions from data obtained from the functioning of an industrial process.

The methodology includes the classification of the data splitting the variables that describe the state of the process in three groups: the variables that describe the input of the process, the variables that describe the state of the process, and the variables that describe the control actions taken.

2.2.3.5.1.1 Parts of the automaton

The automaton has to reflect the different situations the process has had along time. But, if we want to know what has induced these changes, we have to consider not only the situations but also their causes. In the process that is going to be modelled, two different causes exist: the input of the process and the actions taken by the process operator. These terms are easily translated to automata terminology:

Nodes: Each node of the automaton represents a class of the operational situations of the process.

Transitions: Each transition connecting two nodes represents a change from one situation to another. It is labelled with the input of the process and the control actions taken.

2.2.3.5.1.2 Preprocessing the data

The input to the algorithm that builds the automaton is a temporal sequence of situation descriptions. Each description consists on the values of different process attributes measured at a given time that can be either qualitative or quantitative. These descriptors can be split into various groups depending on its nature: input, state and control.

Input: Describe the different properties of the input of the process, that are outside the control scope of the process operator.

State: These are the variables reflecting the operating situation of the process. Among them, there are the variables describing the final product obtained by the process. They measure the performance of the process. The main task of the supervisory system could be maintaining them within a determined range of values.

Control: These are the parameters that the operator can adjust in order to modify the operation of the process. Their modification affects the future evolution of the state (and output) variables.

After these classifications the following partition are obtained:

$$\text{Input } I = \{I_1, I_2, \ldots, I_m\}$$

$$\text{State } S = \{ S_1, S_2, \ldots, S_n \}$$

$$\text{Control } C = \{ C_1, C_2, \ldots, C_p \}$$

That correspond to the different classes obtained by each classification. Now each of the entries that correspond to the snapshot of the different continuous variables of the process is replaced by a 3-tuple that indicates to what input, state and control classes belongs to. So, each entry now has the form:

$$E_t = (I_t, S_t, C_t)$$

And the trace of the whole process is the series of 3-tuples $E_1, E_2, \ldots, E_l$ *. Each of the different classes of S constitutes a node of the automaton.*

Once the nodes are built, the transitions can be computed with an algorithm, able to do this data mining task.

Once the model is available it can be used for example to simulate the future behaviour of the system modelled. The drawback of this automatic knowledge discovering is that for having an accurate model as an output, we will need a great amount of data (training examples) in order to let the algorithm be able to capture all the relations. It also has the drawback of needing to measure enough variables that describe the state *S*, to let discover the algorithm what are those variables that are related with transitions. The more parameters that we have describing the state, the more accurate the algorithm will perform in terms of capturing the causal/temporal dependencies that makes the system transiting from an state to another one.

2.2.3.5.2 Neural networks

Neural networks are named after the cells in the human brain that perform intelligent operations. The brain is made up of billions of neuron cells. Each of these cells is like a tiny computer with extremely limited capabilities; however, connected together, these cells form the most intelligent system known. Neural networks are formed from hundreds or thousands of simulated neurons connected together in much the same way as the brain's neurons.

Just like people, neural networks learn from experience, not from programming. Neural networks are good at pattern recognition, generalization, and trend prediction. They are fast, tolerant of imperfect data, and do not need formulas or rules. Neural networks are trained by repeatedly presenting examples to the network. Each example includes both inputs (information you would use to make a decision) and outputs (the resulting decision, prediction, or response).

Neural Network tries to learn each of the examples in turn, calculating its output based on the inputs provided. If the network output doesn't match the target output, The control system of the neural network, corrects the it by changing its internal connections. This trial-and-error process continues until the network reaches the specified level of accuracy. Once the network is trained and tested, It can accept new input information, and it will produce a prediction.

There are two phases in neural information processing. They are the *learning phase* and the *retrieving phase*. In the training phase, a training data set is used to determine the weight parameters that define the neural model. This trained neural model will be used later in the retrieving phase to process real test patterns and yield classification results.

Retrieving Phase: Various nonlinear systems have been proposed for retrieving desired or stored patterns. The results can be either computed in one shot or updated iteratively based on the retrieving dynamics equations. The final neuron values represent the desired output to be retrieved.

Learning Phase: A salient feature of neural networks is their learning ability. They learn by adaptively updating the synaptic weights that characterize the strength of the connections. The weights are updated according to the information extracted from new training patterns. Usually, the optimal weights are obtained by optimizing (minimizing or maximizing) certain "energy" functions. For example, a popular criterion in supervised learning is to minimize the least-squares-error between the teacher value and the actual output value.

Real-world applications may face two very different kinds of real-time processing requirements. One requires real-time retrieving but off-line training speed. The other demands is both retrieving and training in real-time. These two lead to very different processing speeds, which in turn affect the algorithm and hardware adopted.

Designing a neural network is largely a matter of identifying which data is input, and what you want to predict, assess, classify, or recognize.

Neural networks offer a general-purpose solution to pattern recognition problems. They are able to generalize much better than traditional programs and can run faster. Neural networks are not limited to any set of characters, and can learn to recognize just about anything, even things like tools, mechanical parts, aircraft, and cancerous cells.

2.2.3.5.3 Fuzzy models

Fuzzy models should be built using expert prior knowledge when available. Learning from example data may be used for tuning an existing fuzzy inference system and also for automated model extraction from the data. Bonissone [Bonissone et al., 1999] provides a good survey of building fuzzy inference models combining different methods from classical control theory and soft computing and combining expert knowledge with data based tuning. Also Babuska [Babuska et al. 1999] describes interesting methods for fuzzy modeling also involving data based methods.

Fuzzy inference systems (FIS) may be viewed as a particular model class and also some of the more traditional model types may be fuzzified (fuzzy decision trees, fuzzy clustering and fuzzy SOM).

In fuzzy model extraction, structural learning may involve the selection of the most informative inputs to use (called variable selection) and selection of the order of the system (number of input and output lags). Also the selection of the model type may be involved (Takagi-Sugeno or Mamdani inference model), as well as the identification of the number of rules to use in the mapping. This has to do with fixing the granularity of the fuzzy partition of the input space (selecting the number and type of membership func-tions).

In fuzzy model extraction, parameter learning involves the tuning of the membership function parameters for the inputs and the consequents. Also the other structural parameters could be tuned. Parameters that are linearly related to the output may be optimally estimated by least-squares methods (LSE). One has to use non-linear optimization methods (like neural networks) for estimating non-linear mappings.

2.2.3.5.3.1 Takagi and Sugeno's Fuzzy Model

The fuzzy model suggested by Takagi and Sugeno [Takagi et al. 1985] represents a mathematical tool, which is used to build a fuzzy model of a system. A fuzzy model of a non-linear system consists of a set of implication rules, which are used to express control statements. An implication rule contains fuzzy variables with unimodal membership functions. Since such membership functions are linguistically understandable, the fuzzy variables are also called linguistic variables. Takagi and Sugeno's fuzzy model approximates a nonlinear system with a combination of several linear systems by decomposing the input space into several subspaces and representing the input/output relationship, in each subspace, with a linear equation.

2.2.3.5.3.2 Sugeno and Yasukawa Model

The fuzzy model suggested by Sugeno and Yasukawa [Sugeno et al. 1993] consists of rules whose consequences are represented by linguistic variables which, effectively, results in the Mamdani model. This model is more intuitive than the one proposed in [Takagi et al. 1985] and it is easier to implement.

2.2.3.5.3.3 Pros and Cons

Takagi and Sugeno's model can express a highly nonlinear functional relation using small number of fuzzy rules. However, the complexity of its identification procedures made it difficult to be used. In Sugeno and Yasukawa's model, the identification algorithm is simple. However, because the model uses singletons as consequent parts, it requires many fuzzy rules and its capability of system description is poor.

2.2.3.5.4 Neuro Fuzzy models

It is generally agreed that fuzzy inference systems (FIS) provide a useful way of representing human knowledge in a fairly readable way in form of fuzzy inference rules. FIS rules are also capable of representing inexact knowledge and to reason with such knowledge in a theoretically sound way. However, the tuning of FIS proves to be challenging, as in a nontrivial FIS there are quite a few parameters to modify (typically the member-ship function parameters). It would be useful to be able to create or tune a FIS based on a training data set of input values and the desired target outputs. One of the ideas has been to apply the learning abilities available with the neural network architectures to the tuning of FIS.

In the neural network domain supervised learning task is often solved using a feed-forward layered network structure with simple processing units organized in layers. The nodes in each of the layers are typically fully connected with those of the neighbouring layers. For each of the nodes there are typically only few adjustable parameters like the weights from each of the neighbours and a bias weight. The network is adjusted to a set of learning data (inputs and outputs) by feeding the inputs into the system, propagating the evidence through the network and by calculating the difference from the desired target. Then the parameters are adjusted by gradient descent optimization performed by a special error back-propagation algorithm.

2.2.3.5.5 Support Vector Machines

Support Vector Machines are currently receiving a lot of attention for modelling and classification problems, because of their good empirical performance as well as a firm theoretical background.

Historically, classifiers have been determined by choosing a structure, and then selecting a parameter estimation algorithm used to optimize some cost function. The structure chosen fixes the best achievable generalization error, while the parameter estimation algorithm optimizes the cost function with respect to the empirical risk.

There are a number of problems with this approach, however. These include:

The model structure needs to be selected in some manner. If this is not done correctly, then even with zero empirical risk, it is still possible to have a large generalization error [Vapnik 1995].

If we wish to avoid the problem of overfitting, as indicated by the above problem, by choosing a smaller model size or order, then it may be difficult to fit the training data (and hence minimize the empirical risk).

Determining a suitable learning algorithm for minimizing the empirical risk may still be quite difficult. It may be very hard or impossible to guarantee that the correct set of parameters.

The support vector method is a recently developed technique which is designed for efficient multidimensional function approximation. The basic idea of support vector machines (SVMs) is to determine a classifier or regression machine which minimizes the empirical risk (that is, the training set error) and the confidence interval (which corresponds to the generalization or test set error) [Vapnik1995].

In SVMs, the idea is to fix the empirical risk associated with an architecture and then to use a method to minimize the generalization error. The primary advantage of SVMs as adaptive models for binary classification and regression is that they provide a classifier with minimal VC dimension which implies low expected probability of generalization errors. SVMs can be used to classify linearly separable data and nonlinearly separable data. They can be used as nonlinear classifiers and regression machines by mapping the input space to a high dimensional feature space. In this high dimensional feature space, linear classification can be performed.

In the last few years, a significant amount of research has been performed in SVMs. Learning algorithms and training methods are examined in. Methods for determining the data to use in support vector methods have been considered in [Scholkopt et al. 1995]. Decision rules have been considered in [3Burges et al. 1995]. Applications of support vector machines to speaker identification are considered in [Schmidt 1996].

Support vector machines have been shown to have a relationship with other recent nonlinear classification and modelling techniques such as: radial basis function networks, sparse approximation, PCA and regularization. Support vector machines have been used to choose radial basis function centers .

The key to understanding SVMs is to see how they introduce optimal hyperplanes to separate classes of data in the classifiers.

2.2.3.5.6 Causal Probabilistic networks

A causal probabilistic network, or **Bayesian Network**, is a dependency model defined over an acyclic directed graph, where nodes represent variables and arcs describe dependencies relations (cause-effect) among variables. If a node with a variable A is father of a node (immediate parent) with a variable B, therefore it is considered that B is a direct effect of A. These dependencies are typically quantified in each node with the conditional probabilities distribution associated to this node in respect with variables in parent nodes.

In this way, in a causal network we usually have at the same time a qualitative component, and also a numerical component to represent knowledge: The qualitative describes dependence/independence relations among the variables involved in the problem. The quantitative (or numeric) quantifies these relations using probabilities, or in general using uncertainty measures.

The idea of using graphical representations for probabilistic information can be traced to the geneticist Sewal Wright, who developed the method of *path analysis* "as an aid in the biometric analysis of certain classes of data." The method came under severe attack and was shunned by statisticians during the first halft of the 20^{th} century, until it was discovered by economists, psychologists and sociologists

Bayesian networks, concretely, overcome some of the problems of **Markov networks,** a previous probabilistic model. The main weakness of Markov networks is their inability to represent induced and non-transitive dependencies; two independent variables will be directly connected by an edge, merely because some other variable depends on both. As a result, many useful independencies go unrepresented in the network. To overcome this deficiency, Bayesian networks use the richer language of *directed* graphs, where the directions of the arrows permit us to distinguish genuine dependencies from spurious dependencies induced by hypothetical observations.

2.2.3.6 <u>Implementing MBR</u>

As in other fields, Object-oriented programming is particularly suited to device modelling and is very popular in the filed of AI MBR, because of this clear separation of function, structure and state. Codes can be written for instance in SMALTALK, C++ or Java. However topological models of device behaviour can also be represented in the form of first order predicate classic logic and written in PROLOG, COMMON LISP.

Whatever the language adopted is, a core challenge of MBR is to find out the best modelling language as possible, that is the type of objects, the way causal relationships are expressed, especially temporal relationships, that will produce the most comprehensive, efficient and of broad use system. [Brusoni, Console et al. 1996] gives a good idea of the nature of the problem.

2.2.3.7 <u>Modelling issues</u>

This section recalls briefly the basic difficulties and limitations of modelling when developing an MBR system, in the general case.

2.2.3.7.1 Fidelity (accuracy) and precision

Two major issues of modelling are:

- Fidelity which is another name for accuracy: it refers to the correctness of the statements of representation of the system.
- Precision: refers to the level of details. It can be numerical precision : 15 digits instead of 10 …

"The sun will rise tomorrow morning and set in the evening" has fidelity, but little precision.

The statement: “there are 1,456,876,456 molecules of air are in a tiny bottle”, has great precision but is nor accurate.

Different tasks require different precision and accuracy. For example, basic 2D chemical molecule structure is enough to determine the chemical composition of a molecule, but a 3D representation is necessary to identify stereo-isomers Representations that are correct but imprecise often lead to incorrect predictions.

There is a trade-off between accuracy and precision. For computer model there is also a trade of with efficiency. An increase in precision requires a decrease in efficiency. A typical solution to overcome that latter problem is the adoption of a hierarchical approach. Instead of one very detailed model, a suit of progressively more and more precise models is develop to refine let’s say the diagnostic from blocks to individual components.

2.2.3.7.2 Numerical and misconception errors

If modelling requires some necessary approximations it is at the risk of introducing errors. Errors are roughly of two types:

1. Numerical errors, when the digital word constraints (lets day numbers represented with 13 or 14 digits) put a limit on the precision and/or accuracy of the model.
2. Misconception errors, when, for instance, the law of physics are violated by an unconsidered approximation.

Unfortunately, there is not such thing like a methodology giving advice to model all kind of possible systems that would help avoiding the trouble of making errors. Nevertheless, some recipes are available.

For continuous systems, bond-graphs is for instance a popular proven method which helps avoiding misconceptions errors. Object-Oriented modelling is also a modern and powerful technique. Qualitative Physics may be sometimes of a great help too for physical systems.

Other methods, including for other type of systems, are: discrete-event modelling, automata, Petri net, or Qualitative Reasoning.

2.2.3.7.3 Model domain of validity

Indeed, before the model can be of any use, it is necessary to understand what is its domain of validity. Often it is very difficult to reproduce all conditions in which the system could fail. Thus it is not practical to check out in every possible situation if the model reproduction or detection of the failure is correct (In fact this is why model are used instead of fault trees). Global methodologies to help determine the domain validity of the model and that would apply for every possible system, are still under study.

2.2.4 Principles of Model-Based Reasoning

2.2.4.1 Definition

Model-Based Reasoning (MBR) is the symbolic processing of an explicit representation of the internal workings of a system in order to predict, simulate and explain the resultant behaviour of the system from the structure, causality and behaviour of its components. Qualitative models aim to capture the fundamental aspects of a system or mechanism, while suppressing much of the detail. Methods such as abstraction and approximation are often used to build models based on symbolic rather than numeric quantity spaces. Such models are based on a sound domain theory that provides a systematic, consistent and complete knowledge base for the aspects of the system being modelled.

The main differences in emphasis between these models and the conventional models used in science and engineering is the incorporation of explanation structures and the requirement for the system structure to be explicitly modelled. Such models often turn out to exhibit much better structural similarities with the conceptual basis of the real system.

2.2.4.2 Reasoning techniques

The basic paradigm of Model-Based Reasoning lies in the interaction of observation and prediction. It consists in comparing the model results with some observations of the corresponding system. It involves making a series of experiments with the model until you finally identify what is your system made-up with, or what is going wrong in your system, or what is the best design of your system, or …. The next figure can help to understand how this process can be carried on.

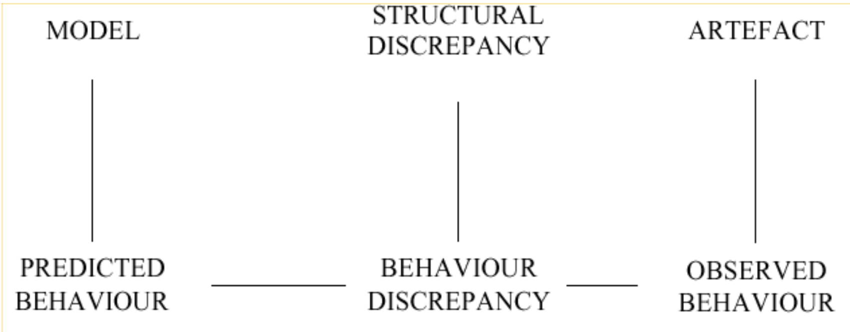


Figure 1. Model-Based Reasoning paradigm : Model-Artifact Differences [Kleer and Williams 1992]

To predict behaviour, different reasoning techniques can be used depending on the values that the variables of the model can take. Following this approach we define Quantitative and Qualitative reasoning.

Quantitative reasoning used models whose variables take numerical values. This is the basis of much of the work of the Fault Detection and Isolation (FDI) community [Patton et al, 1989]. It is possible to use numerical models to predict the expected values of observed variables over time, and thus to detect when deviations from expected values occur due to faults. This kind of model-based monitoring is important for real-time systems. It provides a good way of implementing problem identification, but gives little help for the other areas of diagnosis, where FDI researches use other techniques such as rule-based systems.

There are several reasons why it is harder to use numerical models for fault localization and fault identification. One reason is that many such models are of the processes occurring in the domain, rather than component-based models, and so they have the same limitations as other process-based models. A second reason limiting the diagnostic usefulness of numerical models is that exact values of parameters are no longer available when a fault occurs, and so it is not possible to perform simulations of failure situations. In practice, this means that where numerical models are used in diagnosis, they are only used for problem identification, and other methods are used to decide what to do about the problem that has been identified.

Qualitative models can complement the use of numerical simulation, as they are less effective at detecting that there is a problem, but better at assigning blame to specific components.

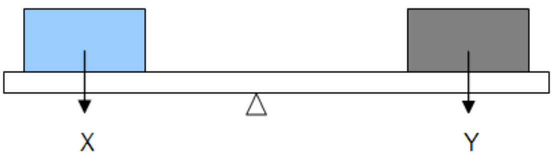


Figure 2. Qualitative reasoning about weights on a balance

Exact values are not necessary when reasoning about many everyday situations. For example, take the problem of deciding what happens with two weights, x and y, one each side of a balance, and the same distance from the pivot, as shown in Figure 7. It would be possible to calculate the answer for each possible pair of values of x and y. There are, however, only three possible outcomes, and they depend on whether the value x-y is less than, greater than, or equal to zero. Zero is a threshold or landmark value for x-y. Any other changes in the values of x and y will not affect the outcome. A change in outcome only occurs when x-y moves from a negative value to zero or from zero to a positive value. Qualitative reasoning involves identifying such critical values of variables as landmark values, and then combining qualitative values to determine the outcome of a situation.

Qualitative arithmetic and a qualitative calculus have been developed for combining and reasoning about qualitative variables that can only take the values +, - and 0. Cohn's survey article (1989) gives a reasonable introduction to the main strands of qualitative reasoning. As Cohn observes:

"The crucial point about a qualitative quantity space is that it is a finite, discrete space which is much more amenable to reasoning about than the underlying continuous quantitative space. In particular, sets of equations can be solved by a finite number of guesses if required, and the number of possible states and behaviours is also finite (unless dynamic landmark creation is allowed)".

One of the great advantages of qualitative reasoning is that there are only a finite number of possible situations to investigate. This type of reasoning is also very useful when exact values are either not significant for the outcome of the simulation (as is the case in much failure mode effects analysis work) or not available (often the case in diagnosis). A related disadvantage is that the results of the simulation are often ambiguous.

The main advantages of the qualitative reasoning approach over numerical simulations are:

- A qualitative simulation can be performed in cases where a numerical simulation would be impossible because of incomplete or only qualitative information being available in the domain. For example, reasoning about the effects of a leak in a pipe is possible, even though the amount of liquid list is not known.
- A qualitative simulation can be performed much more rapidly than the comparable set of numerical simulations (covering the same set of values), and needs much less computing power. This is because of the comparative simplicity of the model.
- The symbolic representation of a qualitative model can closely match the structure of the device being modelled. This makes it easier for qualitative systems to produce explanations and justifications at an appropriate level for human comprehension than is the case with numerical models of a domain.

2.2.4.3 MBR Architectures

The architectures described in this section are designed to solve the following problem: A system and its model are given. The model is described in terms of its structure and components, the system by means of its variables and their values. The observed data of the system differ from predictions provided by the model. It is assumed that the difference is due to a malfunction of some system components. The goal is to find a minimal set of model components whose abnormal function explains the difference. This set is called a diagnosis for the system.

This section focuses in the techniques used in model-based diagnosis reasoning that are a modification of the identification/control paradigm known in control theory. As depicted in "non-diagnostic tasks" sub-section, although they are diagnosis techniques, they can be used for other tasks different from diagnosis.

2.2.4.3.1 Architecture with parallel model

Architecture with a parallel model is a well-known paradigm of control theory. The block diagram that describes it, is shown in Figure 3.

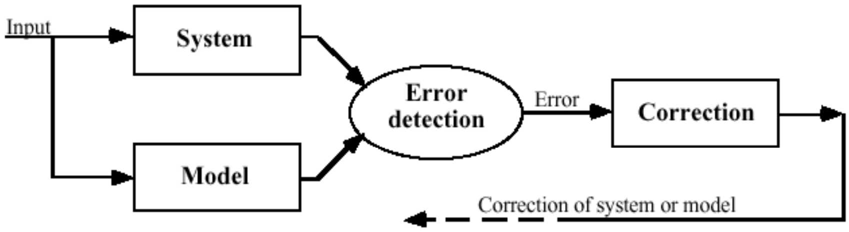


Figure 3. Architecture with a parallel model

A system and its model are defined in terms of the concepts introduced in section 2.2.3. Both the system and the model have the same input values. The responses of the system are observed or measured while variables of the model are calculated. The difference between corresponding variables is an error which is evaluated and adjustments are made to minimize it. Different tasks result from the choice of the adjustable block. In principle, there are two possibilities: the model is adjusted in accordance with system responses or the system is adjusted in accordance with model predictions. The former task is model-based identification (diagnosis), the latter is model-based control. In both cases the corrections are error-driven.

At the beginning of the current section, the model-based diagnostic problem was defined as follows: By adjusting the model find a minimal set of its components whose abnormal behaviour explains the error. This set is called the diagnosis. Now, providing that the presupposition of structural isomorphism holds, the diagnosis describes the faulty components of the system. The structural coincidence is crucial, since we adjust components of the model and by means of this isomorphism we judge the components of the system. Without this assumption, for example, it would be possible to diagnose a full adder built from 2 XOR gates, 2 AND gates and 1 OR gate by means of a model constructed from NAND gates and eventually find a faulty NAND gate of the system although there is none.

2.2.4.3.2 Criterion of minimum error (GDE, Reiter)

Models are adjusted in various ways. The basic GDE relies on the assumption-based truth maintenance algorithm. One part of the model is adjustable since only involved components are assumed to function correctly. Any of these assumptions may be withdrawn whenever necessary. The rest of the model is fixed, as it is built from models of correct behaviour of components and cannot be modified. The component description is taken as the assumptions of the ATMS (see [de Kleer 86]). Similarly, Reiter's diagnostic theory splits the model into two parts: the adjustable one with abnormality predicates included in formulae and the other one without them. The inconsistency is removed by disallowing some formulae of the variable part of the model from participating in predictions. Thus, in an extreme case, the variable part of the model can be completely cut off. The model is adequate if its fixed part is consistent with the system. In general, the part of the model consistent with the system is smaller than the system itself and the mapping from the model to the system is a monomorphism (injective homomorphism). The unassigned (by the monomorphismm) part of the system is supposed to be faulty. The diagnosis with the model of correct behaviour of the system is shown in Figure 2.

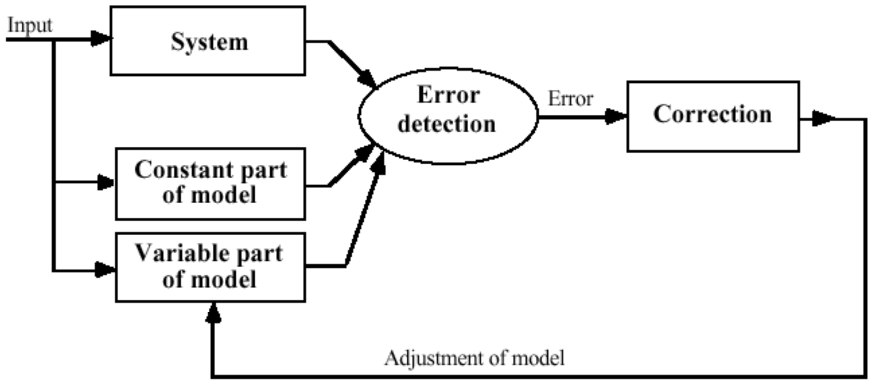


Figure 4. Model-based diagnosis with a model of correct behaviour.

2.2.4.3.3 Criterion of maximum coincidence

The fault modelling techniques make use of a different scenario. The constant part of the model remains the same but instead of gradually removing formulae of the variable part they are rather replaced by suitable fault models. This approach is used by GDE+, Sherlock and to some extent also by NOSTRUM. Since models of NOSTRUM are derived from first principles the same model includes both correct and faulty behaviour. However this advantage is paid for by an increased complexity.

The failure of the system is assumed to be produced by faulty components rather than errors in their interconnection. Fault models have the same structure as the model of the correct behaviour, they are just built from descriptions of faulty components.. The structure fault model is composed of generative systems representing a selection of some properly functioning and some faulty components. We assume that if a fault model is found which is consistent with responses (data) of the system the components of the system obey the same generative laws as the corresponding components of the model. We extrapolate again from properties of the model to properties of the system. Faulty models are of the same size (measured by the number of participating components) as the system, the mapping between the system and the chosen fault model is an isomorphism. The diagnosis with fault models is shown in Figure 3.

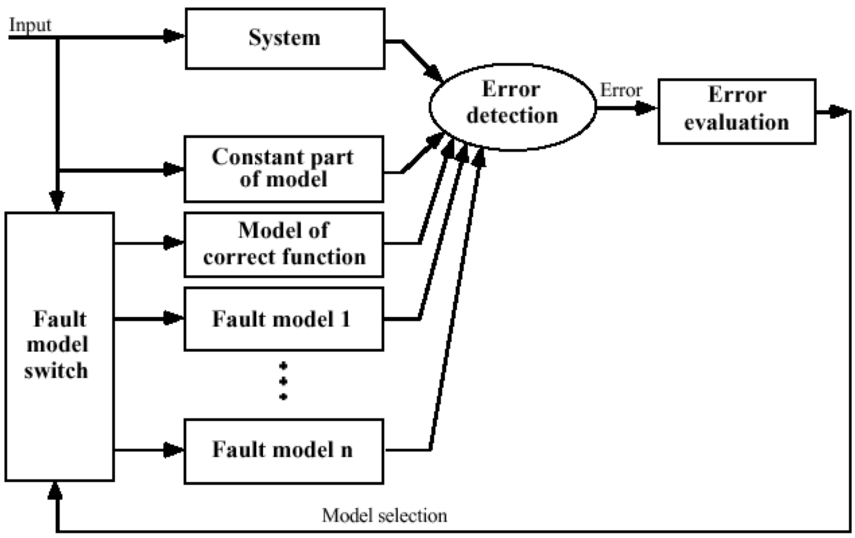


Figure 5. Diagnosis with fault models.

The architecture of diagnosis with fault models is sketched only schematically in Figure 3. In fact the complete fault model is composed of the constant and the variable parts, the latter being built from a set of fault models or from the model of correct function. The fault model switch selects an appropriate variable part of the model. The selection algorithm depends on a particular method. The fault models are mutually exclusive, the overall model of the system includes the constant part and just one of them.

2.2.4.3.4 Non-diagnostic tasks

A similar model-based approach can be used to solve tasks of non-diagnostic nature. The algorithms remain unchanged; the only difference consists in the interpretation of the basic scheme shown in Figure 1.

The diagnostic methods described so far have assumed an error-free observation channel. The data provided by the system are supposed to be measured without any error and differences between them and predictions calculated from the model are ascribed to faulty components. However it is quite possible to commit an observation error. The system may function properly and the inconsistency between the data and the predictions may be caused by corrupted measurements. In this case the task is to validate data and to eliminate inconsistent values. The problem can be solved by a similar technique with the adjustable part being now related to the system. This task is schematically shown in Figure 4.

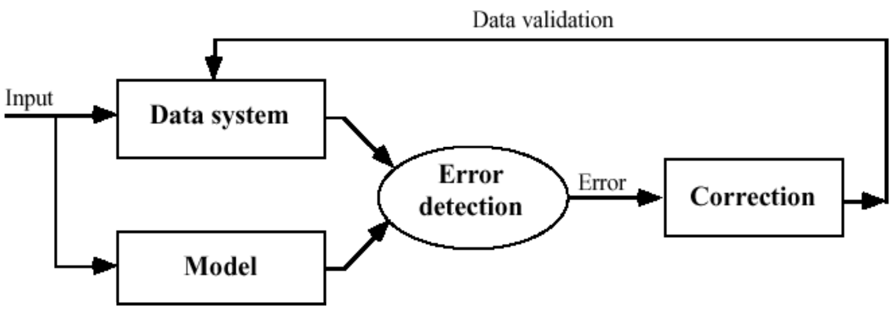


Figure 6. Data validation

The data validation tasks can be solved both by the basic GDE and by Reiter's diagnostic theory although these methods have not been originally designed to do so. GDE just associates the assumptions (of correctness) with data rather than with components of model and then withdraws them if they are proved to produce inconsistencies. Reiter's diagnostic theory will introduce new constants of the language, say m_1, m_2, ..., m_M for M suspicious measurements. Measurement m_i is represented in terms of the formula ~AB(m_i) -> Var-i = Val-j, which means: "Unless the measurement m_i is abnormal the variable Var-i has value Val-j". Otherwise both algorithms proceed normally. This scheme makes it possible to diagnose both erroneous measurements and faulty components simultaneously since it does not make any distinction in the representation of potentially erroneous parts.

Differences between responses of a system and its model can be used to control the system. This is a fundamental idea of feedback control. Also this kind of task can be incorporated into the basic scheme shown in Figure 1. The model provides wanted values of relevant variables and the system is modified in order to achieve them. As in diagnostic tasks, the system is split into two parts: constant and variable. In control theory the variable part is usually called the controller and it is designed to enable an effective control of the rest of the system. The control with a parallel model is shown in Figure 5.

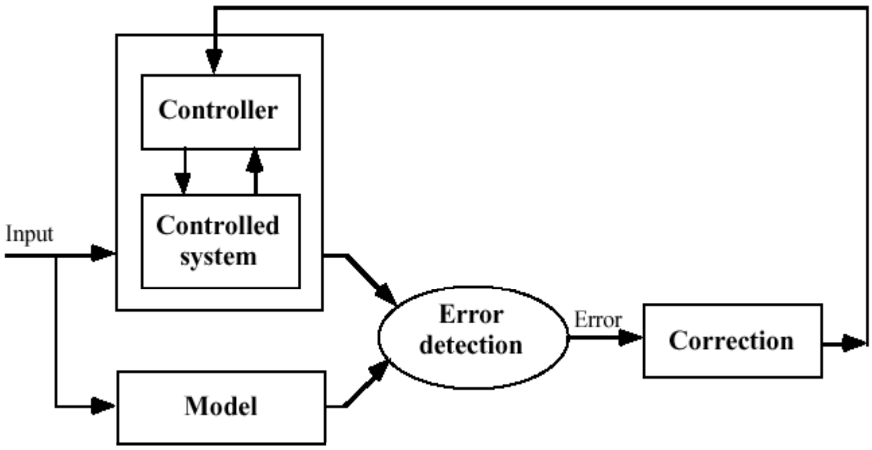


Figure 7. Control with a parallel model.

There is a wide range of control tasks that can be eventually solved in this way. Since the goal is merely to achieve the same responses, the structure of the model need not necessarily coincide with that of the system. The controller defines the class of models which the system can emulate.

The tasks introduced in this section belong to two distinct groups: control and diagnosis. However these can be solved by virtually the same technique. There is an obvious duality between diagnosis and control, the same problem can be interpreted either as a control or as diagnosis. The major part of the support tools including the complete methodology for model building can be used for both groups.

2.3 REVIEW OF EXISTING CBR / MBR APPLICATIONS

2.3.1 CBR Applications

CBR has been used in a broad range of application domains to capture and organise past experience and to learn how to solve new situations from previous past solutions. For example:

- Planning - e.g. PRECEDENTS [Oxman & Voß, 1996], CAPLAN/CBC [Veloso *et al*. ,1996], CHEF [Hammond, 1989]
- Design - e.g. NIRMANI [Perera & Watson, 1996], JULIA [Hinrichs, 1992]
- Classification - e.g. PROTOS [Bareiss, 1989]
- Diagnosis - e.g. CASEY [Koton, 1989]
- Understanding and analysis - e.g. (AQUA [Ram, 1993; Ram & Hunter, 1992]
- Interpretation - e.g. HYPO [Ashley, 1990]
- Troubleshooting detection - e.g. CASIOPÉE, LADI [Lenz et al., 1996]
- Explanation - e.g. SWALE [Kass & Leake, 1988])

CBR provides an adequate framework to cope with continuous domains, where a large amount of valuable new experiences are generated in a continuous way. For example:

- As an assistant for conceptual internetwork design - CIDA [Joh, 1997]
- In planning and execution monitoring in traffic management in public telephone networks - NETTRAC [Brandau et al., 1991].

In Environmental Sciences CBR has been applied in different areas with different goals. For example:

- In information retrieval from large historical meteorological databases - [Jones and Roydhouse, 1995].
- In optimisation of sequence operations for the design of wastewater treatment systems - [Krovvidy and Wee, 1993]

- In supervisory systems for diagnosing and controlling Wastewater Treatment Plants (WWTP) systems - [Sànchez-Marrè *et al*., 1999; Sànchez-Marrè *et al*., 1997a].
- In decision support systems for planning forest fire fighting - [Avesani *et al*., 2000].
- In case-based prediction for rangeland pest management advisories - [Branting et al., 1997].
- In case-based design for process engineering - [Surma and Brauschweig, 1996].

2.3.1.1 EC Fifth Framework - IST projects

In the different IST programmes, projects that use CBR as the core technology have been numerous and a flavour of its the variety is shown in this section:

- **DIETORECS: Intelligent Recommendation for Tourist Destination Decision Making (2001-2004)** The main objective of this project is to develop and validate an interactive and "conversational" recommendation system for destination decision-making in the tourism sector.
- **INSPIRE: Intelligent Support for People Oriented Process Re-Engineering and Change Management (2000-2002)** The goal of this project is to business process re-engineering (BPR) support toolset. At the core of the tool, generic formal process models are being used to represent process data. An implementation planner, built on a CBR engine, is being developed to support the change process.
- **KDNET: European Knowledge Discovery NETwork of excellence (2002-2004).** The objective of KDNet is to fill the current gap between knowledge discovery and data mining. Research, industry and public sector organisations participate in KDNet to jointly shape the European knowledge discovery community, giving it a coherent and highly visible profile. KDNET will integrate sub-communities from database theory, statistics, machine learning, case-based reasoning and neighbouring disciplines such as bio-informatics, economics, and geographic information systems.
- **LIAISE: Local Intelligent Agent as Informed Sales Expert (2000-2002).** This project aims to produce a commercial tool to aid in the configuration and quotation of complex highly configurable multivendor systems along the whole systems value chain. Mechanisms of competition, negotiation and co-operation are supported by the LIAISE architecture, implementing the new paradigm of value federation. This system will incorporate artificial intelligence in the form of rule-based and case-based reasoning.
- **PATTERNS: Patterns To Adopt Knowledge Based Solutions To Software Management Problems (2001-2003)** - Provides an internetworked software application that will allow users to obtain a rapid solution to a context based problem. In addition to modelling the Knowledge Centre domains of management practices, this software solution is based on integration of three different technologies: CBR, Intelligent Agents (IA) and Natural Language Processing (NLP).
- **VIP ADVISOR: Virtual, independent advisor for personal insurance and finance risk management (2002-2003)** is developing a virtual, interactive personal insurance and finance assistant. The knowledge based virtual assistant uses CBR technology.

2.3.1.2 EC Fourth Framework - Esprit projects

- **INRECA: An Integrated Platform for *Reasoning* from Cases (1992-1995) -** The goal of this CBR project was to compare and contrast two approaches to problem-solving using complex real-world applications in biology (medical diagnosis from case records, identification of marine sponges) and technical maintenance (fault diagnosis in aircraft). INRECA resulted in a generic CBR platform for diagnosis and two market-oriented systems focusing on a pharmaceutical industry application and technical maintenance.
- **CHARADE: Combining Human Assessment and Reasoning Aids for Decision-Making in Environmental Emergencies (1992-1995) -** the CHARADE architecture included general

modules built up from a set of basic reasoning and presentation tools for use in different application domains, mainly by adding specific knowledge bases

- **APPLICUS: Trial applications of CBR for customer support (1995-1996).** In this project, CBR was used to: improve after-sale support (telephone hotline) with help-desk software; develop diagnosis and fault analysis decision support systems; update regularly troubleshooting manuals from observed faults, as well as the capture and reuse the experience of the most talented maintenance specialists, enabling the transfer of expertise amongst personnel and the building of a corporate memory.
- **CASTING: Case-Based Reasoning: Stimulation of Industrial Usage (1995-1996).** The objectives of this project revolved around the promotion of CBR, creating awareness in the industrial sector on its potential and on how it may be implemented and exploited in practice.

- **INRECA II: Information and knowledge re-engineering for reasoning from cases (1996-1999)** The central aim of INRECA II was to develop a methodology and a set of tools for building systems that reason from past experience - that is case based reasoning. The resultant methodology was published in book form by Springer [ISBN 3-540-66182-4] 'Developing Industrial Case-Based Reasoning Applications'.
- **BRIDGE: Development of a real-time intelligent diagnosis tool for a large technical applications (1996-1999)** The project had the specific goal to develop a generic fault diagnosis tool for large technical systems. The principles of CBR being used to declare and maintain a unique formulation of the malfunctioning of the entire application.

2.3.2 MBR Applications

The survey undertaken in the RIMSAT project indicates that the methods, techniques, and systems developed are mature enough for industrial applications, at least for selected domains and tasks. Commercial tools for model-based diagnosis tools are starting to appear on the market - for example, IDEA and RODON. In the following sub-sections we show the most important applications of Model-Based Reasoning in European projects and in other environments:

2.3.2.1 EC Fifth Framework - IST projects

- **CogVISys: Cognitive Vision Systems (2001-2004)** The central goal of this project is to build a vision system that can be used in a wide variety of fields and that is re-usable by introducing self-adaptation at the level of perception. CogViSys aims at developing a virtual commentator, which is able to translate visual information into a textual description.
- **MONET 2: Network of Excellence in Model Based Systems and Qualitative Reasoning (2002-2004)** This project has evolved from the Esprit project MONET (see the following section). Its aim is to provide a framework for research co-operation and integration that will promote research in the field of model based and qualitative reasoning and encourage technology transfer.
- **PICK: Tools for Process Improvements Based on Corporate Knowledge Management. (2000 - 2002)** The objective of PICK is to develop two innovative methods and tools for effective management of corporate knowledge needed to support main process improvement (PI) steps, specifically for manufacturing processes. PICK is utilising three different types of reasoning systems, including MBR and CBR.

2.3.2.2 EC Fourth Framework - Esprit projects

- **ARTIST: Advanced Reasoning Tool for Model-Based Diagnosis of Industrial Systems (1990-1993).** The context of ARTIST centred on the increasing complex technological processes and sophisticated equipment that must be supplemented by advanced automation systems to ensure efficient, safe and cost-effective production, whilst guaranteeing high plant availability by minimising down-time. The aim of ARTIST was to develop tools that will allow the diagnosis from observation of the system variables the causes of a failure and, possibly, the earlier detect developing faults.

- **TIGER: Real-Time Situation Assessment of Dynamic, Hard-to-Measure Systems (1992-1995).** This project resulted in a software system and toolkit to provide continuous on-line monitoring and diagnosis. The TIGER system features an artificial intelligence qualitative simulation and model-based diagnosis module, as well as real-time data links to gas-turbine control system and a real-time, rule-based expert system capable of identifying causes of faults very quickly within a guaranteed response time.
- **TIMELY**: Time-Constrained Integrated Management of Large-Scale Systems (1993-1996) TIMELY developed a complete management system for transportation networks that integrates the basic tasks of detection, diagnosis and remedial action within an environment for handling time-constrained reasoning.
- **MONET: Model based and qualitative reasoning systems network (1997-2000)** to provide a long-term framework for research integration and co-operation that will co-ordinate European research in MB & QR systems, and promote technology transfer into industry, open to all those who can contribute to technological progress in MB & QR systems.

2.3.2.3 Model-Based Diagnosis Applications

Today, model based diagnosis shows both a strong theoretical foundation in terms of logical theories [Reiter, 1987], [de Kleer-Mackworth-Reiter, 1990], [Console-Torasso, 1991], [Struss-Dressler, 1989], [Dressler-Struss 1992, 1994], [Dague-Deves-Raiman, 1988], [Tatar, 1994], [Tatar-Iwanowski, 1994] and considerable work on industrial problems, such as diagnosis of power networks [Beschta et al., 1993], [Struss, 1992, 1993], analogue circuits [Dague et al., 1987, 1991], diagnosis and failure modes and effects analysis of automotive subsystems [Price et al., 1992], (PROMOTEX). The General Diagnostic Engine (GDE), is a general domain independent architecture for implementing model-based diagnostic systems. It is a standard that provides the basic computational substrate of large number implementations.

In many situations models are easier to build and maintain than heuristic approaches (do not require experience), and are task independent (the model can be used for many tasks). However, even in devices domains where MBR industrial systems are widely used, the modelling task can be very complex. The main issue is the availability and generation of models, especially for complex manufacturing and assembly processes. Consequently, the model-based approach can only be applied in a limited number of cases.

A more complete view of the most relevant MBR systems can be seen in more details in the next subsections.

2.3.2.3.1 General Diagnosis Engine (GDE)

The General Diagnostic Engine (GDE) system described in [de Kleer-Williams, 1987] makes use of the Assumption-based Truth Maintenance System to find multiple faults of a diagnosed system. The model is represented as a constraint network built from mutually interconnected functional blocks. The blocks describe the *correct* behaviour of individual components by means of relations - constraints among their variables. The interconnections correspond to the coupling among components – some components share common variables. The model thus represents the correct behaviour of the device. Some variables of the model have their values assigned as a result of observations or measurements on the system. If enough data is gathered other variables of the model are calculated assuming that the components used in the computation behave in accordance with their constraints. If the calculated values differ from the observed ones the conflict is discovered and the diagnostic process may start. The GDE consists of three major parts: a problem-solver, an ATMS and a predictor.

The problem solver calculates new values of variables and looks for conflicts between the calculated values and the observed ones. This module has been implemented in the GDE as a standard constraint propagator. The control strategy guarantees that the constraints are evaluated in accordance with an increasing number of assumptions involved. Thus the conflicts are minimal the derivation of diagnoses is simplified.

Inference steps are communicated to the ATMS, which carries out most of the GDE diagnostic work. The problem solver generates *assumptions* which represent assertions that a given component is functioning correctly, i.e. in accordance with its constraint. Assumptions are also processed by the ATMS. Conflicts denote logical inconsistencies which are represented as the minimal disjunctions of inconsistent assumptions (see [de Kleer, 1986]) - the ATMS *nogood*s. The diagnosis is a set of violated assumptions which have produced nogoods. Given n assumptions there are 2 n possible diagnoses starting from the empty “Nothing is faulty” up to the n-tuple “All components are faulty”. The diagnoses are ordered into an environment lattice in terms of subset-superset relation. The diagnosis for a problem is a set of assumptions which subsumes all discovered nogoods (minimal conflicts). Diagnoses are calculated from nogoods as minimal elements of the environment lattice. By removing all inconsistent environments the ATMS “cuts off” the parts of the model that might produce inconsistencies.

The third part of the GDE is the predictor which calculates the most promising future measurement by evaluating the entropy of the model. This procedure is not relevant to our main interest and thus will not be discussed.

The philosophy of the GDE is based on the presupposition that the difference between the correct behaviour of the component and the measured one implies that the component is faulty. This presupposition is generalized to the conclusion that the model of correct behaviour is sufficient for diagnosing. This issue will be discussed in detail later.

2.3.2.3.2 Reiter’s theory of diagnosis

An approach similar to the GDE is described in [Reiter, 1987] . There are however several differences: the method makes use of a different, more formal characterization of the problem, the search for diagnoses is not incremental, and measurements to be taken are not suggested.

The language used for problem representation is the first order predicate calculus (FOPC) with equality. The model is described in terms of first order sentences. The constants of the language correspond to components of the model. Some components may be faulty. Their normal or abnormal behaviour is expressed in terms of McCarthy’s AB(c) predicate. AB(c) states that the component c manifests an abnormal behaviour. Only the components that have included the AB predicate in their definition can be diagnosed as faulty. Typically the sentence with the AB predicate is written as A(c) & ~AB(c) -> C(c) which means that if the component has property A and is not abnormal then it has also property C. If the A is the name of the component then the formula says that if c is a normal A it behaves in accordance with C.

For example the multiplier M1 which calculates the product of two numbers is described as follows:

mult(M1) & ~AB(M1) -> output(M1) = input1(M1) * input2(M1).

This notation means: if M1 is a multiplier which is not abnormal, then the output of M1 equals to the product of the input1 of M1 and the input2 of M1. In fact this is just an instance of a general formula

$\forall$(x) (mult(x) & ~AB(x) -> output(x) = input1(x) * input2(x))

which describes correct multipliers. The implication guarantees that if the left-hand side holds so does the right-hand side.

Alternatively it is possible to use a more restrictive formula replacing the implication by equivalence. The AB predicate serves as a kind of switch. If the x is *abnormal* i.e. it does not manifest the correct behaviour the formula is "cut off". For formulae with implications it means that we cannot say anything about the relation between the output and the inputs. Even stronger conclusions can be drawn for formulae with the equivalence: the output of M1 is not equal to the product of its inputs.

The model is represented as a set SD (SD stands for system description in [Reiter, 1987]) of sentences which describe the correct behaviour of components and a set COMPONENTS = {c_1, c_2 ... c_n } of all components which are supposed to take part in some diagnosis (i.e. which participate in some AB(c) formula). Only these components can be diagnosed as faulty, the others must be correct. Hence the SD splits into two parts: the formulae with the AB predicate and the rest.

An observed behaviour of the diagnosed device is described also as a set of FOPC sentences denoted as OBS. Similarly as in the case of GDE the diagnostic reasoning is invoked by a conflict between the values of observed variables (OBS) and their expected values calculated by means of the model. Diagnoses are computed from conflict sets CS defined as subsets of COMPONENTS for which $SD \cup OBS \cup \{AB(c_i), c_i \in CS\}$ is inconsistent. In another words, for given observation OBS the members of the conflict set are those components which cannot be simultaneously normal. A conflict set is said to be minimal if none of its subsets is a conflict set. Nogoods calculated by the ATMS are minimal conflict sets. The diagnosis is a minimal set of components which logically implies all conflict sets.

Reiter's diagnostic procedure has two parts: conflict generation and diagnosis evaluation. The conflicts are inconsistencies between the model and OBS. They are discovered by a theorem prover. Apart from theoretic difficulties (inconsistency is undecidable for a general FOPC language) the theorem proving is computationally expensive. Reiter carefully controls the strategy of theorem proving in order to increase the efficiency by avoiding futile calculations. However the control is restricted to the computation of diagnoses for a given set of observations OBS. As soon as a new measurement comes, i.e. the OBS is modified, the previous diagnosis is not exploited, and the diagnostic inferences start again from the scratch. This way of reasoning is counter-intuitive, an incremental refinement of diagnoses would be preferable.

There was an error in the original Reiter's algorithm, the corrections are in [Greiner, Smith et al., 1989].

2.3.2.3.3 Fault modelling - GDE+

Diagnostic reasoning which takes into account only models of correct behaviour proved to be inadequate in many situations. The reason is that these models do not necessarily capture all important domain-specific knowledge and therefore logically sound inferences do not guarantee the correctness of the diagnosis when it is interpreted in terms of the real world problems.

We shall demonstrate how the GDE might arrive at quite preposterous results by means of a simplified example from [Struss-Dressler, 1989]. The device with 3 bulbs and a battery is shown in Figure 8.

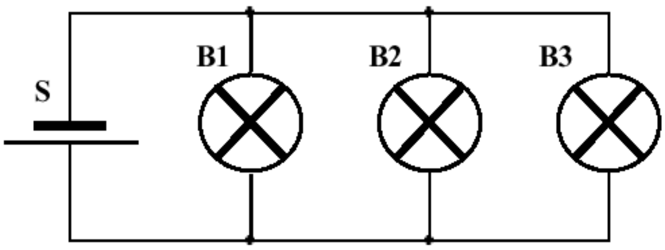


Figure 8. A circuit with 3 bulbs and a battery

The bulbs B1 and B2 are not lit while the B3 is. For simplicity possible faults of wiring will be neglected. The only acceptable diagnosis says that the bulbs B1 and B2 are broken. GDE does derive this diagnosis but, in addition, it offers another one, logically equivalent, which states that S and B3 are broken. This latter diagnosis asserts that the battery S is broken and that is why B1 and B2 are not lit and B3 is broken because it is lit although the battery is broken and there is no voltage. The model of correct behaviour is not sufficient to capture all knowledge. In [Struss-

Dressler, 1989] a solution is presented which supplies additional information about possible faults. An extended ATMS can cope with models of both faulty and correct behaviour.

2.3.2.3.4 Diagnostic reasoning as a mixture of skills - NOSTRUM

The idea of fault models is also exploited in [Nuttall, 1990] where a diagnostic tool called NOSTRUM is described. This program goes one step further than those described above. The diagnostic process is understood as a mixture of various skills which all contribute to the process of diagnosis formation. The most relevant skills are: fault recognition, symptom recognition, tracing, hypothesizing - testing and finally taking repair actions.

The model is derived from the representation of individual components and their mutual coupling. Each component is described by a constraint network which captures its *operating principle,* i.e. the basic laws governing its function. The operating principles are the same both for a component functioning correctly and for a faulty one, the difference being only in a setting of some parameters (a bulb is faulty if its resistance is not within some range). Thus models of faulty behaviour are included automatically. Since the operating principles have close relations to first principles of the component the fault model can be derived although it has not been explicitly defined beforehand. If an observation is made which does not match the prediction various diagnoses are hypothesized and the world of possible solution splits into subworlds. The hypotheses are propagated, their consequences evaluated and a discriminative test is suggested. The results of the test produce new symptoms and the refinement of diagnoses goes on. The hypothesizing and testing is an example of abductive reasoning. Although the method is essentially based on a deep knowledge representation NOSTRUM makes use of any kind of knowledge which might improve the diagnostic process.

2.3.2.3.5 Sherlock

Yet another modification of the GDE which incorporates behavioural models is described in [de Kleer-Williams, 1989]. The task is solved by means of an extended GDE, however the problem is represented in terms of a language akin to that of Reiter's diagnostic theory. Components are considered as being in one of a set of modes each mode having its own description. The mode in Sherlock is a model for the particular behaviour. One of the modes corresponds to the correct behaviour. The others are faulty or just different modes. The modes are mutually exclusive, the component may take just one mode at time. The following example (from [de Kleer-Williams, 1989]) shows the model of a logical inverter with 3 possible modes G (good), S1 (stuck to 1) and S0 (stuck to 0).

$$
\begin{aligned}
&(\forall x)\ \{invertor(x) \text{ -> } [\\
&\quad [G(x) \text{ -> } (input\ (x)=0 \equiv output(x) = 1 \vee input(x)=1 \equiv output(x)=0] \ \&\\
&\quad [S1(x) \text{ -> } output(x) = 1] \ \&\\
&\quad [S0(x) \text{ -> } output(x) = 0]]\}
\end{aligned}
$$

The formula can be instantiated for any concrete invertor. Therefore, if x is an invertor then it satisfies one of three possible models: correct behaviour, the one for the output stuck to 1, and the one for the output stuck to 0. The modes are distinguished by predicates G(x), S1(x) and S0(x).

The diagnosis is a set of all components with associated modes so that the model is consistent with observations. Unlike GDE and Reiter's theory the descriptions of components producing inconsistency are not eliminated, instead they take a mode consistent with the rest of the model. The faulty components are easily recognized since they have been associated with faulty modes. In order to always arrive at a solution the model of each component (the formula above) is completed by a catch-all mode for unspecified errors.

As a consequence of the assumption that all components participate, each one with several models, the diagnostic strategy leads to combinatorial explosion. Indeed, providing that the number of components is n and the number of modes per component is k, there are n k possible assignments. Sherlock controls this problem by means of a probabilistic analysis which eliminates diagnoses with lower probabilities as soon as possible.

2.3.2.3.6 MDS

MDS is a model-based tool that supports several engineering tasks related to the analysis of a system in case of failures, such as: on-board/off-board diagnostic development, safety analysis, diagnosis' analysis, development of test procedures for end-of-line testing. MDS has been developed at DaimlerChrysler research since 1994 [Mauss, May et al, 2000].The overall idea of MDS is to support all the product development cycle, from early design to manufacturing and diagnostic support. The main characteristics of MDS are:

- The algorithm is based on the GDE architecture.
- Prediction is based on local propagation of values restrictions through a constraint network.
- Offer the possibility to define action models for each components. Action can be control action or observation action. Actions are associated with state-dependent cost, (i.e. a number that reflect how easy or expensive is to carry out an action) and with preconditions (reflecting the situation in which the respective ation is allowed to be performed).
- Broad applicability: rather than focusing on a restricted domain for behavioural descriptions, such as finite-domain (qualitative) models, linear models, non-linear models, or finite-state machines, MDS has been designed to be able to deal with all this domains. The pay-off (admitted by its designers) is the limited processing capability of the models and, sometimes, a limited declarativity of models.

MDS has been successfully tested on several industrial applications.

2.3.2.4 <u>Model-based reasoning applications in complex domains</u>

William J.Long et al. [W.J.Long & S.Naimi & M.J.Cristello & R.Adusumilli] have constructed a system for assisting in the diagnosis and orientation of medical tasks in including MBR. They deal with uncertainty, capturing it in causal relationships with a probability network. The human body is a highly integrated functioning unit, medical domains are difficult to define 'cleanly'. Consequently, it is necessary to identify the range and depth of the medical domain.

Other systems that model scenarios with different degrees of complexity also exist. The main handicap of these MBR systems is the exponential complexity of the inference algorithms over the complexity and interconnection of the elements of the domain. Even the Rich Probabilistic Relational Models [Segal & Taskar & Gasch & Friedman & Koller], [Getoor & Friedman & Koller & Pfeffer] that offer a more expressive way of modelling a complex domain, and more efficient inference algorithms, are affected by this complexity boundary.

2.3.3 CBR + MBR Applications

The use of Case-Based Reasoning (CBR) plays a significant role in many relevant tasks like diagnostic problem solving [Kolodner, 1987] or planning [Hammond, 1989], since it can mimic (at some extent) the capability of human experts in solving a new case by retrieving similar cases solved in the past and by suitably adapting them to the situation at hand. The suitability of CBR to solve complex problems has been widely discussed in the last few years and two basic possibilities emerged:

- Precedent Case-Based Reasoning where previous solutions to cases similar to the current one are used as a justification for the solution of the current case with almost no adaptation (e.g. legal reasoning [Ashley et al., 1987]);
- Case-Based Problem Solving where retrieved solutions to previous similar cases need to be adapted to fit the current situation (e.g. planing, design, diagnosis, etc...).

While a pure Case-Based approach seems quite adequate in the first case, research in the second area led to several proposals where case-based reasoning is combined with other problem solving approaches like rule-based [Bonissone et al., 1990; Rissland et al, 1989], prototypical [Torasso et al. 1992] and model-based reasoning (MBR) [Goel, 1989; Koton, 1989; Jang, 1993]. In domains where a precise domain theory is available and analytical methods exist for solving the problem, the advantage of using CBR (possibly in conjunction with other methods) could seem less obvious with respect to domains where the domain theory is very partial and weak. However, CBR can still provide advantages when the precise computation of a solution is very complex; this is often the case when pure model-based approaches are used, so this kind of integration has been studied for tasks like design [Goel, 1989], planning [Jones, 1992] and diagnosis [Koton, 1989, Jang, 1993]. The possibility of organizing and retrieving cases from a dynamic memory can also be viewed as an attempt to bridge the gap between associational and model-based systems. It is known that associational systems are fast but they lack precision because of their heuristic nature; on the contrary, model-based systems are more reliable but less efficient (For this reason, several approaches have been proposed in order to either combine associational and model-based systems [David et al., 1993] or directly focus model-based reasoning [Kleer, 1991; Console et al., 1991]). The identification of previously solved problems can be a useful tool for improving the performance of a model-based system by using experience in problem solving.

There are two basic possibilities in combining CBR and MBR:

- CBR is the main problem solving method and MBR is just used to provide guidance to it (for instance for judging similarity as in [Pews et al., 1993]);
- CBR is used to focus MBR in the attempt to augment the basic mechanisms of MBR by taking experience into account [Koton, 1989].

The Early CBR system Protos [Porter-90] uses a multi-relational model for reasoning when producing explanations to support its case retrieval step. Here each relation is assigned a number corresponding to its explanation strength. In the Creek system [Aamodt-94], this has been taken further by defining symbolic relation types corresponding to particular degrees of certainty (e.g. always-causes, typically-causes, sometimes-causes). Another early system integrating case-based and model-based reasoning is Casey [Koton-89]. An example of a later system is [Branting-99]. Unlike logic-based methods, all the above example methods capture uncertain knowledge, and are able to handle incompleteness. It may also be worthwhile to look at recent methods where description logic is used for model-based reasoning with CBR [e.g. Arcos-96]. Some of them may have incorporated separate inference methods – outside first-order logic - to be able to express and reason with uncertain and incomplete knowledge (at least possibilities to do this has been discussed).

2.3.3.1.1 CARMA

L. Karl Branting and J.D.Hastings [1994] have designed a system for rangeland pest management that uses model-based matching and adaptation to integrate case-based reasoning with model-based reasoning for prediction in rangeland ecosystems. Using both technologies together they have created a system that predicts the behaviour of complex and poorly understood physical systems. The technique created - 'model-based case adaptation' [L.K.Branting-J.D.Hastings, 1995] - serves to increase predictive accuracy in incomplete systems, either because a complete state description for such systems cannot be determined or because the number and type of interactions between systems elements are poorly understood; and/or in systems where lack sufficient data for effective use of empirical methods, such as case-based reasoning, decision-tree induction, or statistical techniques.

CARMA makes use of model-based adaptation as a technique for integrating case-based reasoning with model-based reasoning in domains in which neither technique is individually sufficient for accurate prediction.

The central component of their system is coupled a CBR and MBR system able to “estimate the proportion of available forage that will be consumed by grasshoppers using case-based and model-based reasoning”.

CARMA is based on “a CBR that elicite information to infer the relevant features of a new case. When the relevant case features have been determined, CARMA can use a causal model to assist case-based reasoning in four different ways: case factoring, temporal projection, feature adaptation; and critical period adjustment.”

Both search similarities and adaptation weights are adjusted by the causal model (the MBR).

CBR and CSP integration

CSP (Constraint Satisfaction Problem) can be regarded as a modelling technique. Depending on the problem to solve, it is a more appropriate technique than other classic modelling techniques for MBR.

[Sqalli-Freuder, 1998] are concerned with diagnosing interoperability of network protocols. They propose to integrate a CSP model and a CBR to perform automatic tests suites and relieve the experts from their usual burden when:

- testing a large amount of data to find out where there is a mismatch between what is expected and what is observed
- wasting time in diagnosing problems that his colleagues has diagnosed before or in solving problems that are very similar to old problems solved.

They argue that they choose a combination of CSP and CBR because although been a simple system, its knowledge was incomplete in two ways:

- interactions with the external world not always planned;
- models may contain mistakes due to human limitations.

They give interesting recommendations on using their technique, depending on the level of complexity of the system and on completeness available for the knowledge about it.

[Sqalli, Purvis et al. 1999] proposes also a survey on similar applications (COMPOSER, JULIA, CHARADE, CADSYN, CADRE, CBAR, IDIOM … etc …).

[Purvis, 1998] proposes a criteria to help deciding whether or not it is interesting to combine CSP and CBR.

2.3.3.1.2 EPAION

EPAION is an exemplary case of utilisation of CBR to extend the ability of MBR to deal with external influences that cannot be captured by the model.

While trying to develop an efficient CBR system for aircraft **in-flight troubleshootin**g, [Karamouzis-Feyock, 1992] produced an early work in the area of MBR/CBR coupling. They focused uniquely on physical systems. Their application intended more precisely to produce an in-flight fault diagnosis and prognosis of aviation subsystems, particularly jet engines.

According to them, a physical system is fundamentally component based, with eventually sub-systems. Components are connected together in order to achieve certain functions. When an accident occurs, for example a fan in the left engine, problems can propagate downstream and be captured by a prognostic model of the engine. More unpredictable problems such as flying fragments from the faulty fan damaging the right engine won’t be captured by a model, but may be captured by a CBR. The model can then further determine what will the consequence of flying fragments entering the right engine be.

Moreover, the model might be a combination of deep-rules and shallow rules reasoning. The physical description and behaviour of the device is recorded in the deep-rules while cause and effect relationships, more difficult to capture in the model (like the flying fragments), are recorded in the form of shallow rules. Such that, when a case is learned by the CBR (after being revised by a human expert) it can also teach the model by adding-up corresponding shallow rules. To do this, the CBR must contain, besides the traditional list of symptoms, some temporal information indicating the consequences of the original defect.

In addition, the model can help the CBR to identify failures that look different at a superficial level but have identical causes.

2.3.3.1.3 ADAPTER

ADAPTER is a typical example of an application using CBR to “focus” (accelerate) MBR getting the best of both solutions: the rapidity of reaction of the CBR without loosing the completeness and reliability of the MBR.

The MBR used is apparently an hybrid system, that is based on a mixture of consistency-based and abductive method, but the main component, probably at the origin of the system, is of abductive type.

Base on the following figure, [Portinale, Torasso et al. 1994] give the following description of his system:

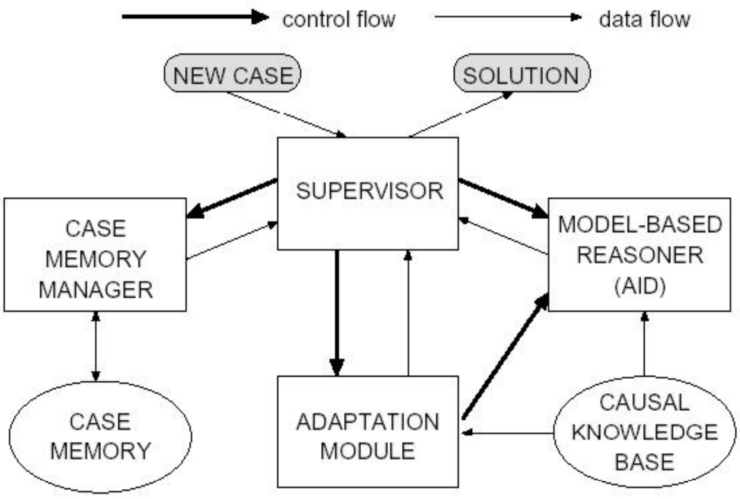


Figure 9. Integrated System Architecture.

In the diagnostic system we can identify the following basic components (see also figure 9):

- a case memory with an E-mop-based organization of cases [Kolodner, 1984] where each case represents a diagnostic problem already solved;
- a knowledge base, represented through a causal model identifying the faulty behaviour of the system to be diagnosed;
- a supervisor module controlling the activation of other modules of the integrated system; -- a module able to store and retrieve cases from the case memory and to evaluate the degree of match between the current case to be solved and the retrieved ones (case memory manager);
- a Model-Based Reasoner (the Aid system) able to perform diagnostic reasoning on the causal model in the form of abduction with consistency constraints [Console-Torasso, 1991];
- a module performing adaptation on retrieved solutions and able to invoke the model-based reasoner if adaptation criteria fail to provide a solution.

The diagnostic system, when presented with a new case, first invokes the case- based reasoner in order to retrieve the most similar cases solved in the past and then it tries to use the solutions of retrieved cases in order to focus the model- based reasoner in the search for the actual solution. The supervisor is intended to manage the above control strategy (notice that also a pure use of the Aid system is possible if the supervisor decides to by-pass the case-based component). The emphasis of the paper is on the adaptation strategies working on the solutions retrieved from the case memory, in order to avoid the computation from scratch of the solution of the current case.

2.3.3.1.4 CHEF

CHEF application uses MBR to improve the performance of the CBR system within the cooking domain.

CHEF [Birnbaum, Collins et al. 1991] core system is a CBR system that retrieves and adapts old cooking recipes to create new ones. For example, if anyone wants to cook chicken and water chestnuts, the system will substitute “chicken” and “water chestnuts” with “chicken” and “snow peas”, and propose an adapted recipe based on the known recipe for chicken and snow peas.

Standalone, the CBR does not perform optimally. Depending on the type of vegetable, it may happen that the recipe to cook beef and green peas would have been a better choice for a stating point to create a recipe for chicken and water chestnuts, because it would have required less adaptation, even if it requires more. However, as the recipe for chicken and snow peas requires less substitution, it is retrieved first.

Improving the CBR can be achieved by:

- detecting a sub-optimal performance occurred.
- determine the cause of the problem.
- effect a repair.

CHEF approach uses explicit models of the planning and plan execution processes to generate expectations about the desired performance of the planning system. The failure of such an expectation therefore indicates a need to modify the system in some way so that it can approximate the desired behaviour more closely.

2.3.3.1.5 MOCAS

This work represents an early effort to make use of modelling knowledge to help CBR systems. In this case, the model is used for similarity assessment and adaptation.

[Bergmann, Pews et al. 1994] investigated the impact of using explanation-based similarity for case-based retrieval and adaptation in the frame of a diagnostic (MOCAS) and a planning (PARIS) application.

The idea is to take into account the fact that domain knowledge is easier to acquire than problem solving knowledge and can be sufficient to derive explanation of the correctness of a case. Such explanations, constructed on several levels of abstraction, can be employed as the basis for similarity assessment as well as for adaptation for solution refinement.

Problem solving knowledge describes the process of problem solving in terms of steps (i. e. basic inference or subtasks as in KADS) that should be executed to (efficiently) derive a solution. Unlike problem solving knowledge, domain knowledge consists of descriptions of the “elements” that are available to construct a problem solution, together with the knowledge about the interaction of these elements within the solution to a problem.

This approach is different from derivational analogy in two ways:

- only domain knowledge is required.
- only the correctness of the solution (that the solution solves the problem) is checked.

2.3.3.1.6 CBR + CSP + adaptation

[Purvis-Pu, 1995; Purvis, 1998] is another work in the area of coupling strategies between CBR and CSP focusing, in this case, on adaptation strategies.

The idea is to merge several cases retrieved since information to solve the current problem can come from several cases. A difficulty usually encountered for this process is that merging is a very difficult task because the results would not be consistent with the problem to solve. To overcome this problem, the authors propose a methodology that formalises the adaptation process using constraint satisfaction techniques. Each existing case is represented and stored in the case base as a solution to a primitive constraint satisfaction problem (CSP) with additional knowledge that facilitates the retrieval and matching processes. The solutions of the primitive CSPs are combined into a globally consistent solution in an ad-hoc way.

2.4 DECISION SUPPORT SYSTEMS BASED ON CBR / MBR - FOR EMERGENCY EVENTS

The goal of the RIMSAT project is to create an interactive, rapid-response decision support solution for those involved in incident management in emergency/safety-critical situations. The 'input' for the system will come from elicited knowledge, know-how and other sources of information both inside and outside an organisation. The main technologies that will be used inside system centre on case and model based reasoning. The 'output' from the system will be a set of action possibilities and/or consequential information to support the end users in the decision(s) they take.

2.4.1 State of the Art

2.4.1.1 Decision Support Systems

A computerized DSS is in fact a software product, often with a complicated structure, sometimes equivalent to a customized software environment: however, usually it can be run on diversified hardware configurations. By its architecture we thus understand not the hardware, but the software composition and organization of the system.

Historically, the architecture of a DSS evolved together with the definitions of a DSS. The concept of a DSS was first used in the late 1970s in apposition to earlier data processing systems: a DSS was supposed to have more functions than simple data processing or a management information system (MIS). This in turn was prompted by the development of interactive computing [Sprage, 1983; Adriole, 1989]. A corresponding early definition of a DSS given by Spragy stressed that it is an interactive computer-based system designed to help decision makers use data and models to solve unstructured problems. However, interactive computing is today taken for granted, not only in DSSs; even for problems that seem to be well structured, an approach encouraging learning about the problem rather than solving it might be preferable. Thus, only the issue of helping decision makers maintained its importance, although both the interpretation of what a decision maker is and what it means to help her/him also evolve over time.

2.4.1.1.1 Basic components of a DSS

Early DSSs were primarily data oriented, by soon it was acknowledged that there should be more possibilities of evaluating alternatives or decision options and even suggesting “best” decisions.

The most straightforward way of model-based decision support provides the opportunity to evaluate scenarios through some form of simulation. A more sophisticated form suggests decision options or supports a search process for the most attractive option. This can be based on using various analytical decision support tools, including optimization algorithms or solvers, together with models in analytical form (sometimes also called operation research form).

However, there are also other ways of obtaining decision options or alternative solutions. For instance, we can use knowledge bases and rule-base reasoning, or models in logical form together with algorithms of an inference engine type (i.e. software to check the validity of a logical statement). However, in environmental applications we usually encounter some physical or biological basic processes that have reasonable analytical models. In theory, we could convert any analytical model into a logical one by defining sufficiently basic concepts or facts and specifying the necessary logical relations between them; however, the growth of computational complexity related to such a conversion might be prohibitive.

2.4.1.1.2 Developing a DSS

The process of developing a DSS often revolves around five building blocks:

i. Information resource management. In software engineering terms, input data are required for decision analysis and resolutions; output data are generated and presented to decision makers for policy making. Effective management of these data constitutes a first major task of any decision support tool.

ii. Model management. A model is an abstraction of reality whose purpose is to help decision makers focus on the main elements of a problem. Multiple objective optimization under constraints is a classic modelling approach in management science. Qualitative reasoning, expert heuristics, and data mining are alternative methods to formulate decisions. Given a decision problem, the challenge of DSS is to find the best decision method(s) able to suggest a satisfying solutions to policy makers.

iii. Interactive problem solving. Direct interaction between the DSS and its user allows for a more responsive and user-centered view of the problem. A goal DSS is one that provides the right information to the right person at the right time with full transparency. In addition, DSS should provide some cognitive feedback to decision makers by helping them comprehend dynamic changes in the underlying assumptions.

iv. Communications and teamwork support. Decision making, more often than not, involves more than one decision maker and support for communication and coordination is an important dimension of DSS. Support for information exchange, generated organizational memory, group decision and negotiation is an integral component of organizational decision support.

v. DSS as non-human co-workers. In a tightly connected networked world, we postulate a working scenario in which humans will team up with computers as co-workers t optimize execution of business decisions [Negroponte, 1995]. We envision a new social structure that emerges from the interaction of individuals –both humans and non-humans- operating in a goal-oriented environment under rules that place only bounded demands on each individual's information and computational capacity [Bui, 1999].

The immediate value of using these five building blocks is to help the DSS users improve their decision outcomes. DSS should achieve its support mission by lending a hand to its users: More quality input data are expected to provide a more complete assessment of the problem situation and a richer set of decision alternatives. More sophisticated decision algorithms are expected to help decision makers find solutions that could not have been found otherwise. Expansive real-time trade-off analyses and interactive simulations are expected to provide decision makers with further insights. Communications and groups decision supports are expected to increase the chance of finding a shared vision and socially equitable solution. Finally, computerized coordinated DSS workflow should seamlessly enhance the integration of sustainable development at a national or global scale.

Perhaps, a more far-reaching value of using DSS is its ability to improve the way decision makers approach the problem, i.e., new insights into the business, better decisions and faster responses to unexpected situations, and most importantly, a changing consciousness about environmental responsibility. Altogether, DSS should help its users become informed workers in dealing with their information-intensive sustain-centrist tasks.

2.4.1.2 Emergency Situations and Incident Management

Although the RIMSAT solution will be applicable across many different application areas where incident management in emergency or safety critical situations is an operational necessity, the initial prototype system will be trialed and validated by UK fire brigades. The rationale for the choice of fire brigades to test the system is that their modus operandi is both highly complex and safety/life critical. Their ability to operate effectively is totally dependent on the knowledge, know-how and experience of the individuals reacting to an emergency situation. Their knowledge comes from a variety of sources, formal and informal, tacit and explicit. They need to make life/safety critical decisions very quickly, decisions that can have a profound effect on the environment, the economy and the safety of the community.

The ability to capture, manage and elicit the necessary knowledge, information, data and know-how, to make it dynamically available to support on-the-spot decision making is seen as key by the fire brigades involved in RIMSAT as a way to improve their quality of service, their competency and responsiveness, and of enhancing safe working procedures - helping to reduce the considerable risk of injury and death that exists today amongst fire brigade response teams.

> "Fire service operations can be fast moving and highly threatening events. Decisions and actions are made rapidly, without quiet contemplation and sometimes with little solid information. Rightly the whole process is seen as one of dynamic assessment where decisions and actions change with the unfolding circumstances. This can be confusing to the public, perhaps even seen as chaotic. To the fire-fighters this is their workplace. The link between action and survival is real."
>
> from an investigative report by HM Fire Service Inspectorate for Scotland

In terms of decision support and knowledge management, current practice within the UK fire services is that:

- Knowledge is perceived as a reusable source for training and ready to use experience for pragmatic goals - i.e. reactive.
- Acquired knowledge is stored and accessed only when it is perceived that it is needed, resulting in slow transfer of best practices within organisations.
- Acquired knowledge is presented as a 'static' record of what has gone before.
- Acquired knowledge focuses only on practices that are perceived as good or best.

2.4.1.3 Knowledge Management

Knowledge management has only recently emerged as a discipline exercisable in an industrial context. Major corporations 'discovered' that the high level of technology and performance they reached was in fact based on an intangible asset - the set of the pieces of knowledge captured in the memory of their employees. Capturing and capitalising that intangible asset is now viewed as essential to an organisation's measurable performance. This is especially true in the USA, with its short-term view on corporate performance significantly affecting the perceived equity of a company. Huge sums have been and are being spent on implementing knowledge management systems.

Knowledge management is fundamentally socio-cultural i.e. people based, so it is no coincidence that information technology for easily connecting people and information has blossomed at the same time that knowledge is becoming recognised as the most valuable of a company's assets. There is a powerful symbiotic relationship between knowledge management and IT. As information technology has become our personal desktop tool and our link to each other, so we grow to covet even more access to information and people's knowledge. In turn, we demand even better IT tools - ones that become part of the way we work.

Until recently, knowledge management techniques have been used to transform the critical tacit knowledge into valuable and reusable sources of training and ready to use experience for pragmatic goals (to avoid the reproduction of errors, reproduce valuable solutions, reduce development delay and cost, etc.). In this context, the application of IT to KM is traditionally classical in its use - captured knowledge stored in relational databases accessed when required by the user. We can call this a 'reactive' use of corporate knowledge - knowledge available as a source of information when the user determines it is needed.

The next major step forward will be the 'proactive' use of corporate knowledge. The manipulation of multiple sources of knowledge to provide reasoned, anticipatory support for the user. This proactive use of knowledge is the foundation of the RIMSAT project.

The application of artificial intelligence to knowledge management is today at a very early stage in its commercial evolution. However, from a research perspective, various knowledge-based techniques have been considered for knowledge management and reuse activities over the last three decades. For many years, the cost and performance of computing hardware was a major barrier to the practical, commercial adoption of these research results. Even today, despite the widespread availability of inexpensive, powerful personal computers, virtually all real-world diagnosis systems use simple decision trees or rule-based methods, rather than knowledge-based 'reasoned' techniques. With the recent advent of algorithms sufficiently powerful for real-world diagnostic problems the application of knowledge-based techniques to knowledge management is poised to 'take-off' in commercial terms.

2.4.2 Existing Applications and Projects

2.4.2.1 <u>DAI-DEPUR</u>

For the last 10 years, I-R. Roda and others [Sànchez-Marrè, Cortès et al. 1998; Sànchez-Marrè, Cortès et al. 1999] have been involved in the designed of an AI empowered control system for a waste water plant.

The DAI-DEPUR system [Sànchez-Marrè, Cortès et al. 1998; Sànchez-Marrè, Cortès et al. 1999; Cortés, Sànchez-Marrè et al. 2000; Matas 2000], is a DSS designed to automate the control and monitoring of a Wastewater Treatment Plants. It is based on a hybrid system whose central component is a CBR system capable of dealing with continuous data time series. The inputs of the CBR are continuous time series produced by sensors measuring critical parameters of the wastewater plant, and the CBR outputs are as well continuous time series of actuation commands chosen to maintain the plant in the appropriate functioning range. In case something unusual happens, the CBR is doubled by a classical expert system and Model-Based reasoning system that control the coherence of the CBR commands.

The challenge for the CBR system is here to deal with continuous time series such as those produced by dynamic continuous systems. Firstly, an expert system select all relevant data coming from sensors (apparently this step is not mandatory, the system could have as well find by itself which cases are relevant or not). Then, the CBR is used.

A structural CBR has been chosen to insure domain independency. This also eases the use of raw data coming from sensors and allows the connection of the CBR-agent with other modules.

The library is implemented as a prioritised discrimination tree, to match a faster access to cases.

A specific Eixample distance has been implemented to compute similarities. Depending on some weight criteria for attributes, it is computing a qualitative distance or a quantitative one.

The adaptation is based on interpolation of the best similar case. A linear interpolation between the values of the attributes of the situation and the values of the actuation parameters is performed.

As usual for CBR, both successful and unsuccessful cases are used for learning.

“Continuous domains present some added difficulties to the building process of a CBR system.

To decide which element of domain constitute a case. Some regular sampling points have to be carefully chosen with an expert to obtain discrete values in time and space of the continuous system.

- The size of the case library could grow very fast without any really relevant improvement of the system. This is appealing for mechanisms to forgetting and sustained learning mechanisms.
- The case base may have to be frequently reorganized to keep the search fast enough (discrimination tree).”

2.4.2.1.1 Relevance, utility, case ontology.

Two concepts have been introduced to cope with the two latter points:

- To learn only the relevant cases
- To establish a lazy learning algorithm for storing cases in the case library.

Cases are declared relevant if their "distance" with other cases is big enough. If not, they are discarded.

A normality measures is keeping track of actual usefulness of cases. When it reaches 0, cases are considered useless and are "forgotten", that is, the case stored in the case base is not used any more.

Taking all these parameters in considerations an ontology can be defined. It specifies what does the CBR system (learning, forgetting …) do in what situation.

2.4.2.1.2 Lazy learning

Beyond discrimination and decision trees, the lazy learning algorithm will delay the split of braches until it is strictly necessary. It will be done to distinguish some previously similar stored cases from the new case, in order to maintain the discrimination/decision power of the case library. The criteria is based of course on the presence or absence of new attributes but also on the similarity distance Eixample distance.

2.4.2.2 HITERM and A-TEAM projects

The HITERM project, funded in part by the European Community (High-Performance Computing for Technological Risk Management, [Fedra-Winkelbauer, 1999], http://www.ess.co.at/HITERM/). It is a very advanced and complete Emergency Decision Support System that can deal with emergency incidents ranging from industrial accidents, to fires, explosions and evacuations. The suit of simulation tools used includes: source model (release of chemical: pool evaporation model), atmospheric dispersion (multi-puff, multi-layer Eulerian, or Lagrangian approach based on 3D diagnostic wind field model), fire and explosion models, stochastic soil contamination routine, as well as a Monte Carlo routine to estimate risk thresholds. It is built around an hybrid expert system whose "knowledge base domain is shared between forward and backward chaining strategies, so that backward chaining inference can affect the forward chaining rules and vice versa".

"The central focus is the interface between technological risk management and the environment, providing a complex and demanding testing ground. Using distributed parallel computing with massive parallel machines or workstation clusters, the project aims at reaching better-than real time performance for the simulation of accidental releases of hazardous substances into the atmosphere, ground and surface water. Interactive **decision support** for emergency planning, training, and management is the goal of the project."

The HITERM project has been prolonged with the A-TEAM (Advanced Training System for Emergency Management) project : http://www.ess.co.at/A-TEAM/ which is uniquely concerned by improving the learning process in complex technical domains.

2.4.2.3 The ARTEMIS project

The ARTEMIS European project (Application Research testing of Emergency Management Systems, [Hernandez-Serrano, 2000] which has resulted in the designed of two mock-up Emergency DSS systems, one of them dealing with flood emergency and the second one with industrial accidents (e.g. heavy gas dispersion).

It has been build around a conversational model with the tool KSM (Knowledge Structure Manger). The modelling part involves Qualitative Physics.

2.5 APPLICABILITY OF CBR & MBR IN RIMSAT

2.5.1 Criteria for qualifying advantages and drawbacks

Understanding why coupling both techniques can be of any interest requires understanding first which are the drawbacks and advantages of both methods with respects to each others. This naturally will lead us to find cooperative strategies combining advantages of both methods.

A pure technical listing of what brings to CBR or to MBR coupling strategies seemed too restrictive to us. We found necessary to integrate as well users, developers, administrators and top-managers point of views. To make this possible, we gathered information by blocks, each covering one topic, and start to give pointers to the most relevant blocks for each possible point of view. We choose this strategy of presentation because it is very flexible. View points can be added and blocks can be easily completed.

By nature as it is difficult to reconcile some methods qualities, trade-offs must be decided. Some compromises must be agreed by combining both methods (CBR-MBR).

2.5.1.1 <u>Viewpoints</u>

2.5.1.1.1 User view points

Search performances (completeness and unusual failure, fuzzy searches, speed)

Guide the user

Flexibility

Methods limits (traps and erroneous answers, multiple results)

Justification

2.5.1.1.2 Top-manager view point

Re-use

FMEA

Certification

First deployment

Scalability

2.5.1.1.3 Administrators

Learning

Scalability

Modelling aspects (domain knowledge, problem solving knowledge, expert knowledge and device knowledge).

2.5.1.1.4 Developers, Knowledge engineers and people interested in pure technical CBR + MBR issues are

All the concerns already mentioned.

Modelling aspects (domain knowledge, problem solving knowledge, expert knowledge and device knowledge).

Scalability

2.5.1.2 <u>Blocks by alphabetic order</u>

Certification	Guarantying that a given system will always behave in the same way.
MBR	Is a good candidate for certification. It will always retrieve the same failures, even if efficiency (the speed searches are performed) may be tuned by adjusting some first guess probability of failures.
CBR + MBR	Design the CBR system with the help of an MBR system in order to be able to select a rigorous list of symptoms that are necessary and sufficient to rigorously demonstrate, that is support, the results. Checks-up consistency of cases retrieved and adapted with the help of MBR.

First deployment	When starting from scratch a new application in a new domain, for a new system with a new expert
CBR, Induction	Easy and fast. Require only expert knowledge modelling. The expert selects intuitively the relevant symptoms. Arbitrary choices are first made for similarities (with the Knowledge engineer advices), there are tuned by test and try, until results are satisfactory. It can be done in a few days to a few months.
MBR	When nothing exist, MBR first deployment can be a very heavy job. It requires to find out the appropriate domain knowledge and also modelling, with some problem solving algorithms enhancements that make the system efficient. It can take up to several years. However, once the system is completete, can be used in a large class of systems that share the same domain knowledge (all digital circuits).
CBR + MBR	Deploy a CBR first. Move to MBR when ready. Use the real experience (real cases) acquired when using the CBR to check-up the quality of the MBR.

Flexibility	Rather than being forced to provide in a given order symptoms, an expert may prefer to choose freely himself among all possible symptoms which one he will answer first.
CBR	CBR is flexible. The system would perform better if some symptoms are answered first than others but, in any case, the user can start by whatever symptoms he wants and get the same results (if he ends-up filling-up the same list of symptoms with the same values).
MBR	Similarly, MBR is completely flexible.

FMEA	Failure Mode Effect Analysis (FMEA). • Identify the ways in which a process can fail to meet critical customer requirements • Estimate the risk of specific causes with regard to these failures • Evaluate the current control plan for preventing these failures from occurring • Prioritise the actions that would improve the manufacturing process In short, FMEA helps companies to anticipate failures in the manufacturing process and prevents future occurrences.
MBR	Can be used to produce FMEA
CBR	Induction FMEA can be treated as cases for Case based systems.

CBR + MBR	Produce FMEA with MBR during the product cycle. Efficiently retrieve them when needed with CBR during, for example, post-manufacturing phase.

Guide the user	Help the user to reach the conclusion as fast and at the lowest cost as possible. The issues involved: • Save time and money when the user is a beginner. • Save time and money when the system is complex, when it can take time to check-up every possible symptoms, when some measures to identify symptoms are more expensive or time consuming or risky than others.
CBR	CBR differentiates between symptoms. This is one of its strength. If the user is an expert, this is not a problem and he/she may prefer to use CBR rather than Induction because he/she can proceed to his/her search in one step instead of filling-up attributes one at a time. CBR can be combined to induction to improve this.
MBR	MBR starts like CBR with whatever information is available. It gives some candidates results on this basis. And, often propose some best next “probe point” to filled-up in order to reach conclusion, reduce the number of candidates, as fast as possible or/and at the lowest cost as possible. It thus gives more guidance than CBR.
CBR + MBR	Back-up the CBR by using MBR to provide guidance to the user for selecting best probe-point.

Justification	Explanation of an action or a decision, that is, the list of motivations that affected or made this decision or action possible.
CBR	The list of case retrieved with the values of local and global similarities to the query
MBR	A rigorous proof of the conclusion. From the model to observation discrepancy, a cause-effect chain supporting the conclusion. May not be as visual as CBR because the user is supposed to understand the structural and behavioural laws underlying the reasoning (i.e., the law of physics) and to be able to check-out the mathematical demonstration of the solution, that is, to re-do it himself. He can always trust the system that is supposed to deliver only conclusions that are mathematically demonstrated.
CBR + MBR	Design the CBR system with the help of an MBR system in order to be able to select a rigorous list of symptoms that are necessary and sufficient to support the results.

Learning	Knowledge acquisition is a important issue in automatic decision support systems The issues involved: There is often a trade-off between automatic learning and completeness. When a system is automatically and continuously learning, it often means that it does not know everything when it starts to be operational. All case based system fall into that category. As long as their case based is not complete, they are not complete. However, in case of a complex

	system is impossible to design a complete model.
CBR	CBR also learns automatic. In case of structural CBR stronger constraints on the type of cases that can be "learned" have to be considered.
MBR	MBR knows everything from scratch. The model used should be complete. All possible situations can be diagnosed or generated at wish. This can actually be an alternative to a case base that is by nature incomplete. It does not have to acquire knowledge. However, it can make use of first guess probabilities that may be adjusted when confronted with results. MBR may learn through other systems like CBR.

Methods limits – *traps, erroneous answers*	A user may need to be guided but also not to be confused or "induced" to a wrong conclusion. Some systems are more dangerous than others as regard to this problem.
Induction	Pure induction is very sensitive to incomplete queries. In a decision tree, when some information is missing, the search must be stopped. The result is then probabilistic on unsorted cases and can lead to erroneous answers (the most probable result may not be the good one). Dynamic induction can help limiting the problem by making use of all available information but it may stop unfinished if there is not enough descriptors filled-up t reach a leaf. Induction results quality depends also on the completeness (and quality) of the case base used.
CBR	As for induction, CBR quality depends on case base completeness and quality. In case of an unusual query, that does not correspond to a case contained in the case base, a very low similarity values may warn the user that no good solution is available for that particular case. However, this is not guaranteed and it may as well produce a wrong answer.
MBR	MBR is normally delivering only well demonstrated and proved results, supported by a complete model of the system. MBR consistency-based does not differentiate between an unusual or frequent answer.
CBR + MBR	Back-up CBR with MBR in case of complex problems that cannot be properly resolved by CBR. This can start by simply providing both systems to the user and if he/she is not satisfied with CBR results, he can start an MBR based search, that is generally slower but more reliable.

Methods limits – *Multiple results*	In case of diagnoses, it may happen that several failures happen at once. Often systems are strongly limited in case of multiple failures, some of them presupposing just single failure.
CBR	In case of multiple failures, CBR may return several very different cases (one class for each failure). The user, by looking at local similarities should be able to identify this problem.
MBR	In general, MBR systems are limited to the search of a single failure, however, they exist sophisticated systems that include the ability to deal with multiple failure whatever their number is.

CBR + MBR	Back-up CBR with MBR in case of complex problems that cannot be properly resolved by CBR. This can start by simply providing both systems to the user and if he/she is not satisfied with CBR results, he can start an MBR based search, that is generally slower but more reliable. The reverse order in the process can also be found in the literature (first MBR and using CBR as a resource).

Modelling – *Domain knowledge*	All the knowledge necessary that represents facts, relationships, descriptors, objects, functions of the device or the system. The issue involved is the difficulty of the domain knowledge modelling.
CBR and Induction (Case Based Systems)	Choice of a list of relevant symptoms (descriptors, attributes, objects …), either nominal or numerical, mono or multi-valued, integer, real, symbolic (ordered or not), plain text, intervals, taxonomy …. They are often chosen informally by the expert system, and a more rigorous selection may be required. The selection may lead, for instance, to a list of necessary and sufficient elements to demonstrate the conclusion. One of the strength of case based systems, and in particular CBR, is precisely be able to provide reliable results even in case of complex system impossible to model completely.
MBR	Domain knowledge has always to be captured rigorously. MBR tries to model in an expert independent, device independent and even domain independent and, task independent way. Expert independent and device independent modelling is achieved through “first principle” modelling, the laws of physics. For a given domain, for example, analogical electronic systems or mechanical systems, the same modelling language can be used (i.e., laws of Joule). When higher level languages like qualitative physics, that provide general behavioural and structural descriptors, are used, it is said modelling from “second principles”. This is used for instance for digital circuits (AND, OR gates …) Domain independent modelling is still largely an open subject of research. A popular technique consists in relying on Ontologies. Task independent knowledge consists in finding a common modelling language for designing and diagnosing [Mauss, May et al. 2000]. There is a trade-off between domain independent and task independent modelling. Therefore, it is hardly feasible to achieve both goals simultaneously [Fensel, Motta et al. 1997].

Modelling – *Problem Solving Knowledge*	All problem solver systems can be viewed as search engines each one with its own problem solving method and algorithms. There is a strong interaction between the Problem Solving Methods used and the knowledge modelling (Domain knowledge) aspects. Problem solving methods are independent of the domain knowledge.
CBR	The CBR search relies on the similarity heuristics choices. Similarities must be empirically tuned by test and try until desired performances are reached. This is both a weakness and a strength. CBR systems are very flexible but similarities must be adjusted depending on the domain, the expert and the device knowledge.
MBR	One of the objectives of MBR is to decouple PSM and Knowledge

	Modelling. Great effort has been devoted to this subject of research.

Modelling – *Expert Knowledge*	Experts often introduce a bias when modelling. The issue is to poduce systems insensible to that expert modelling bias.
CBR	Experts introduce bias by choosing cases attributes (descriptors, symptoms, etc).
MBR	MBR systems, based on more reliable knowledge like the laws of physics (first principles) or Qualitative Physics are by nature expert independent.

Modelling – *Device Knowledge*	Device, even of close nature (two digital circuits) can introduce different modelling requirements. The issue is to produce systems insensible to device modelling differences. This is important because a given device can be modified during its life time and it should be easy to adapt systems to changes.
CBR	From one device to another, cases structure have in general to be modified.
MBR	MBR are designed to be device independent. It can be as easy as drag and drop components or adjust components parameters to match the new device or the device modifications. Normally there is no impact on the problem solving performances.

Re-use	During the product cycle a lot of documentation and data is produced, and rarely re-used. However, some systems can take advantage of these data to improve the productivity and quality of the product process by: • Reusing data produced by one task (for instance, data produced during design for diagnostic and maintenance) • Facilitating communication between engineers of different skills dealing with different stages of the product process • Reusing data and knowledge from one product to another. • Helping to organise data and select the most relevant one.
CBR	Can reuse data produced during the product cycle to perform another task (for instance diagnostic).
MBR	MBR can achieve the Re-use feature. Models are specified for the design phase and the same model can be used along the product chain.

Scalability	A system under our scope of study is scalable when it is continuously learning, normally to improve its results. The issues involved: • Instant knowledge transfer. For instance, in case of a help-desk application, a new solution is instantly made available to all other co-workers. • No certified regular behaviour can be guaranteed for incremental systems. For a given constant query, results may change with time, as new cases are taken into account.

CBR	CBR is fully scalable. A new case can immediately be incorporated into the case base. Alternatively, it can be chosen to check-it up (i.e., for consistency) by human intervention before re-using it.
MBR	MBR alone is not scalable, the model is supposed to be perfect. The process of developing the model is in itself incremental, but the MBR system cannot be used until the model is finished.

Search performances – *Completeness, unusual failure*	Will the system perform optimally in all cases?
CBR	CBR systems may lead to wrong answers depending on the completeness of their case base. In some conditions, it is impractical to have a complete case base (digital circuits), as this would lead to too many cases.
MBR	MBR does not distinguish between an unusual or frequent answer. Moreover, if the model is accurate enough, unusual failure will be fully diagnosed by MBR.

Search performances - *Fuzzy searches*	Refers to the ability of issuing appropriate results even with an incomplete or imprecise query. Moreover, refers to the resistance of the system against noise.
CBR	CBR is very robust to fuzzy searches, error and noises. All available information is taken into account with no particular preferred order, and a result is returned in all cases. Users can easily verify at view if the result is appropriate or not.
MBR	MBR systems can deal with fuzzy searches starting with whatever available information providing some candidate results on this basis, and refining them afterwards to reduce the number of candidates, as fast as possible and/or at the lowest cost as possible.

Search performances - *speed*	Refers to the time taken to return a result for a given problem to solve, a given query (and system, task, etc). The issue is to reduce the response time.
CBR	Average. Linearly dependent on the number of case.
MBR	The slowest, and probably the least regular. MBR can be slow on some topologies. It can even start infinite lops in case of bad treatment of feedbacks.
CBR + MBR	Accelerate MBR with CBR: focus MBR with CBR [Portinale and Torasso, 1995].

2.5.2 Advantages and drawbacks of CBR and MBR

The previous subsections describe the generic characteristics of CBR and MBR systems from different points of view (user profile) in order to discover the weakness and strengths of each one. As a result of this previous analysis the next subsections describe in a compact way the main advantages and drawbacks of CBR and MBR systems.

2.5.2.1 CBR

2.5.2.1.1 Advantages

- Allows the reasoning engine to propose solutions to problems quickly, avoiding the time necessary to derive those answers from scratch.
- Allows a reasoning engine to propose solutions in domains that are not completely understood. CBR systems can be designed without knowing the physical foundations of the system.
- Gives a means of evaluating solutions where no algorithmic method is available for evaluation.
- Cases make possible to interpret open-ended and ill-defined concepts.
- Cases help the user to focus on important parts of a problem by pointing out what features of a problem are the crucial ones.
- Positive and negative experiences can be incorporated into the case base, thus improving the reasoning engine and avoiding past mistakes.
- CBR systems can be deployed fast and easily as they require only expert knowledge modelling.
- CBR systems can be developed incrementally. The case base can grow while the system is gathering new experiences.

2.5.2.1.2 Disadvantages

- CBR does not pretend to find the right solution to a given problem. Instead it helps the user to find the right solution by presenting the proper cases when they are needed.. If the user is gathering information on the problem, having multiple results can be considered as an advantage.
- It could happen that the parameters which compose the definition of a case are related in such a way that certain values of a parameter alter the relevance of some of the other parameters. In such a situation, the computation of the similarity between cases will be extremely complex.
- “Cold start”. The case base has to include initially enough cases to issue an accurate solution.
- The user has to relay on and interpret the similarity measure in order to get a justification of the solution given by the system.

2.5.2.2 MBR

2.5.2.2.1 Advantages

When the model covers the whole spatial quantity and the domain theory is systematic, consistent and complete over the system being modelled, then an execution algorithm can produce predictions from first principles for each new case - rather than on the basis of limited past experience. This is very important because, with respect to the model, the completeness of the results can be guaranteed. To achieve this completeness usually means that the domain theory must be based on well-formulated principles or scientific laws rather than ad hoc rules or heuristics. Such principles can be applied universally wherever their conditions hold, unlike empirical and experiential knowledge.

This approach has several important benefits:

- **Closer affinity with the real system:** Quantitative and mathematical models are often significantly transformed to become structurally very different from the original system. In MBR the main patterns in the system structure are usually reflected in the model and so interpretation and explanation are much enhanced.
- **Principled approach:** A domain theory with underlying principles provides the reference for model manipulation and reasoning. Thus it is contained and complete.

- **Completeness:** The MBR approach provides for the generation or treatment of all cases within a well-defined framework.
- **Unexpected cases:** It is not necessary to have prior experience of all types of behaviour. Completeness ensures that exceptions do not exist.
- **Better explanations:** The explicit structure of the model and the grounding in a strong domain theory allows explanations to be generated that can justify model behaviour.
- **Aids knowledge acquisition.** The model itself can be examined to verify its correct construction. Domain experts can check the structure and component details and constant use and reuse will enhance the detection and reduction of errors.

2.5.2.2.2 Disadvantages

Model-based reasoning is a relatively new technique and despite intense research activity, there are few full-scale applications in industry or commerce - and as mentioned above, most of them are focused on fault diagnosis. Although there remain many interesting technical and theoretical problems for the research community, the most pressing issues concern technology transfer into the commercial arena. If *MBR* is to realise its full potential the following problems must be solved:

- **Modelling is difficult:** Building principled models for a domain is difficult, time consuming and expensive. Unlike heuristic methods it is not incremental. Consequently, a decision to invest in the technology must often be made without the 'encouragement' of a small prototype or preliminary demonstration. This is an important barrier that slows progress in commercial application development.
- **Model-builders:** Due to the amount of learning and effort involved, model designers would benefit greatly from any available toolsets - unfortunately, very few such toolsets.
- **Reusable libraries are needed:** Component libraries are usually built up as a product of the model construction process. However, they should be designed for reuse in order to save expense and increase reliability. Work on ontologies and component catalogues will ease this problem.
- **Model reuse across tasks:** Models used for design, should also be applicable for diagnosis if the model builder takes account of the task requirements. Unfortunately, models are often seen as tools that assist one particular process and not as a central resource that could support various kinds of reasoning task on a given product or system.
- **Integration with other methods:** When different tools are smoothly integrated into a modelling and simulation environment, then sufficient benefit would be gained to encourage a wider adoption of MBR as the basis for commercial solutions.

2.5.3 Applicability of advantages and drawbacks based on RIMSAT user requirements

As part of the study on current user practice (Workpackage 2) within the UK fire services participating in the RIMSAT project, a 'users wish list' was compiled, based on numerous interviews with senior fire service personnel. A 'mind map' was created (attached to this report as Appendix 1) which summarises all the items on the users' wish list, and which formed the basis for the initial design specifications for the RIMSAT system. Based on the Users' Wish list, and after considerable consultation with the fire services involved in the project, a set of actual user requirements for RIMSAT has been agreed.

RIMSAT will offer 3 types of sessions:

- Incident session will be used during an incident in order to deliver decision support to the user.
- Test and training session will be used offline in order to test the system on specific cases or procedures and to enable the users to get trained on specific aspects of an incident.

- Management session will be used to create and modify the content of RIMSAT (document, cases, models…).

In addition, it has been agreed that RIMSAT will also enable: profile management; document management - for both internal and external documents; case management; model management; data collection; ability to handle contradictory information and time dependent data; decision support tactics.

Based on the three main sessions (incident, training and management), we have assessed the advantages and drawbacks of CBR and MBR for the RIMSAT system. We have categorised the advantages as Essential (E), Useful (U), Nice-to-have but not essential (N). We have categorised the drawbacks as Critical (C); Potentially a problem (P) and non-critical (NC):

2.5.3.1 CBR

Advantages	**Incident**	**Training**	**Management**
Propose solutions to problems quickly	E	U	N
Propose solutions in domains that are not completely understood.	U	E	U
Evaluate solutions where no algorithmic method is available for evaluation.	E	E	E
Interpret open-ended and ill-defined concepts	E	E	E
Indication of critical aspects of a problem	E	U	U
Warm of potential problems that have occurred in the past	E	E	E
Incorporating new experiences (learning)	E	E	E
Fast and easily deployment	E	E	E
Disadvantages			
CBR does not find the right solution to a given problem.	NC **	NC **	NC **
Values of a parameter alter the relevance of some of the other parameters.	C	C	P
“Cold start”	C	C	C
Justification of the solution	P	C	C

** It is important to understand that the RIMSAT system is being designed to offer advice and guidance to the end user, enabling better decision making. It does not attempt to provide a perfect solution.

2.5.3.2 MBR

Advantages	**Incident**	**Training**	**Management**
Close affinity with the real system	E	E	E
Principled approach	E	E	N
Completeness	E	E	E
Unexpected cases	U	U	N
Better explanations	U	E	E

Advantages	**Incident**	**Training**	**Management**
Aids knowledge acquisition	N	E	E
Disadvantages[1]			
Modelling is difficult	NC	P	P
Very few modelling toolsets.	NC	P	P
Reusable libraries are needed	NC	P	NC
Model reuse across tasks	NC	P	NC
Integration with other methods	NC	NC	NC

2.5.4 Discussion about applicability of CBR+MBR in Rimsat

When designing a complex system such as RIMSAT, there are several important decisions to make in order to get the most of the technology and knowledge available. First of all, a decision has to be made for defining the expected output(s) of the system. After having defined *what* do we want, we need to define *how* the system will achieve it. This subsection is a discussion about which are the most interesting outputs that we can get from the designing of an an hybrid architecture including MBR + CBR.

From the beginning, the issue of having a Case Based Reasoner within RIMSAT has been clear because the proposed class of problems has the main characteristics of a successful CBR application that are:

- Broad by shallow domain: there are a number of loosely connected problems that must be dealt with, and they need different kinds of expertise.
- Experience, rather than theory, is the primary source of knowledge.
- Solutions are reusable: when a new problem is seen, it is likely that it can be solved using an old solution.

Model Based Reasoning, in turn, was selected to be used in the project as theoretically it covers aspects that are not solved by CBR and therefore having these both technologies together will make a more complete and helpful system. The main aspect that complements CBR is that MBR does not depend on the human experience in solving problems, instead is strongly related to basic laws of the domain to be modelled. It is in this modelling task where some problems have been found, problems related with the representation. We highlight these three issues:

- Creation/Selection of specific representations useful for the task.
- Possibility and/or opportunity of deriving one representation from another.
- The way different representations are used (and when)

Models are closely related with the information structure. The approach used to model a system is probably very specific to a particular problem and therefore not generally applicable. One reason for this situation is that there are accepted standards for keeping information in data bases, but there are no accepted standards for model building.

Dealing with MBR there are two basic assumptions:

- There is sufficient amount of domain knowledge to be able to write down a ‘deep’ model of the domain.
- The deep model is correct and reasonably complete.

These assumptions have represented an obstacle in the definition of RIMSAT as the domain knowledge owned of the emergencies domain is relatively small in comparison with the wideness of it. However, solutions can be proposed focusing the MBR subsystem in a subpart of the domain.

[1] NB all these disadvantages are from the perspective of the system developer or system administrator, not the end user

Moreover, the second hypothesis is not realistic and some models allow representing incomplete knowledge.

Along the current section, a deep analysis of how MBR and CBR is used in different systems has been the main topic of the research. Also a review of applications that holds these technologies in an hybrid or standalone format has been done. Above we have identified and discussed the most interesting options for creating an hybrid architecture and having useful utilities within RIMSAT framework.

2.5.4.1 Planning

Planning for complex problems with multiple interacting goals and multiple interacting alternative choices is a well-recognised hard problem due to the exponential growth of the search space as a function of with the problem complexity. In contrast planning for simple problem is a rather tractable task [Veloso, 1994].

RIMSAT domain has to be considered as extremely wide, therefore automatic planning in such an environment becomes an intractable problem. However, by reducing the scope of the scenario to deal with, it could be possible to face a planning problem in a concrete subpart of the domain. For planning both techniques used in RIMSAT (CBR and MBR) has been successfully used in the past. In one hand, we find planning with CBR with the remarkable work started by Veloso [Veloso, 1994] and, in the other hand, we have algorithms of planning with MBR. This discipline has been very fruitful in the robotics field (see [Simmons, 1992]).

As it has been said before, CBR has had a clear role in RIMSAT and the type of CBR system that is going to be used does not belong to the family of CBR for planning. Therefore, the remaining option to discuss is using MBR for local planning with a possible toolbox or co-operative integration level with CBR.

Planning with MBR seems at first glance an attractive option for RIMSAT, but analysing it we realise that it is not a realistic option as for issuing a useful plan we would need the highest level of detail in the model. Taking into account the outputs of work package 3, it would be presumptuous to pretend achieving plans when the user expectations are limited to advices that help the incident commanders to face certain situations.

2.5.4.2 Adaptation

Model-based adaptation is a technique in which a domain model is used to assist in adapting solutions associated with past cases to apply to new problems [Goel, 1991]. The purpose of model-based adaptation is to reduce the search associated with a complex model and to improve the performance of a model-based system by using experience in problem solving.

This method was applied in CARMA (Case-based Management Adviser) [L.K.Branting-J.D.Hantings, 1994] in a rangeland pest management domain. In this case, after retrieving the most similar case from a case library of only eight cases, the model-based adaptation method is applied in four ways (factoring cases into subcases, temporal projection, featural adaptation and critical-period adjustment) to give the most accurate rangeland pest management advise to ranchers.

ADAPTER system [L.Portinale-P.Torasso-C.Ortalda-A.Giardino] focuses on adaptation criteria that can be used in a diagnostic system combining case-based and model-based reasoning. The adoption of such adaptation criteria to the retrieved solutions can be viewed as a focusing technique for the model-based inference engine. The process starts retrieving a case from the case base thanks to CBR and MBR is applied for adaptation. In this case, adaptation can occur at different levels:

- Consistency checking: this implies that if consistency is verified, the retrieved solution can be used as a potential solution for the current case; otherwise, inconsistency has to be removed.

- Inconsistency removal: this phase disproves the causal chain leading to the discovered inconsistency, by removing instances of states and/or manifestations from the retrieved solution.
- Explanation construction: this phase builds abductive explanations for entities to be accounted for.

Although the systems in which this method is applied deal with incompleteness and often with lack of sufficient data for effective use of empirical methods, the degree of complexity is lower than RIMSAT's one. Moreover, such adaptations criteria strictly rely on well-defined formal notions of diagnostic problem and diagnostic solution. The fire incident environment has such a degree of diversity that cannot be depicted in a small number of cases nor characterized in an effective model.

2.5.4.3 Consistency checking

ADAPTER system [L.Portinale-P.Torasso-C.Ortalda-A.Giardino] uses MBR in the consistency checking task, that is, the verification to determine whether the characteristics of the input case and the ones retrieved by CBR engine are contradictory or not. In this case, the consistency check is done by comparing the set of ground atoms denoting the value of manifestations that are known to be false in the case under examination with the observations. In case consistency check fails, inconsistency removal mechanism disproves the causal chain leading to the discovered inconsistency, by removing instances of states and/or manifestations from the retrieved solution.

For accomplishing these tasks, a well-defined formal notion of the problem and the solution are needed. As stated before, the model task within the whole RIMSAT domain is an intractable problem and it can only be performed for subparts of this domain. Therefore, the consistency checking and inconsistency removal can be used in a restricted subdomain where a well-defined model could be applied.

2.5.4.4 Retrieval

Wondering what could be the biggest weaknesses of a CBR system like RIMSAT's one we easily identify the similarity assessment stage (in the Retrieval phase of CBR cycle). As it has been depicted in previous sections, when we are dealing in an environment where the relevance of the attributes depends on their values, (i.e. the smoke colour could have no relevance if there is a fire and the fire is in a forest) as far as we are aware, there is no other way for facing this situation than using local similarity functions for performing the similarity assessment. However, this approach has several weaknesses. Basically, there will be a local weight for each parameter's value not taking into account the dependences that may exists between the other parameters' values.

A more sophisticated approach is the work done by [Armengol et al., 2001] where they use local weighting techniques computing the similarity between common features and then normalizing this result by the number of relevant features.

We have proposed a system where the dependences between the attributes can be captured in a model with the expressivity of a simple bayesian network. A simple weighted graph of dependences between all the attributes that define the domain could be used by a similarity computation algorithm in order to compute in an accurate way the similarity between cases. With this method a seamless level of integration between CBR and MBR could be achieved and an accurate computation of similarity can be reached.

The benefits of these methods are related to the easyness of modelling the dependences and in the correct and complete computation of similarity. Moreover, the same model of attribute dependences can be used for other ends enriching the nodes with other kind of information. This model would be constructed manually.

2.5.4.5 Producing explanations

In certain environments having an explanation of how and/or why a problem could have happened is important in order to prevent future repetitions of it. Explanations are also important for diagnosis problems and recent techniques are helping in open domains as medical diagnosis, moving an step forward from the original diagnosis techniques in closed domains as circuit diagnosis.

For producing explanations all the MBR systems use a very detailed causal model. We would like to highlight the work by [Long W. J, 1996] where a system for assisting in the medical diagnosis and the orientation of medical tasks in the concrete domain of hearth failure is explained. Their system perform validation of a large medical knowledge base for diagnostic reasoning using a pseudo-Bayesian network of physiologic causal relations within the *Heart Failure Program* (HFP), that is a computer system which assist the physician by computing differential diagnoses for cases from the findings, represented as detailed graphical physologic causal diagrams justifying each of the hypotheses.

In RIMSAT it can be valuable to have a diagnosis of what could have been the reasons for having reached certain final state. This information could be very valuable in the debriefing process but, again, we face with the issue of the wideness of the domain. As it has been said before, diagnosis tasks require detailed models or in any case enough information for constructing a reliable probability model.

Other systems that use Model Based Reasoning techniques as the system ADAPTER [L.Portinale-P.Torasso-C.Ortalda-A.Giardino], use explanation as one step in the adaptation process, leading the diagnostic system to a more precise answer but also to a greater computational effort.

2.5.4.6 Monitoring

Monitoring is another area where MBR can be usefully applied. The aim is to detect problems as they develop distinguishing between reasonable and important changes from the standard situation. The requirements to deal with this task is to continuously receive data from the system monitorized thanks to sensors and to modelize the behaviour of the system to detect possible inconsistencies to react in some predefined way. The characteristics of the RIMSAT system do not fit with the description above, as the emergency incident can appear in whatever imaginable location and it is unrealistic to count with sensors in all this possible venues. Moreover, the diversity of scenarios where the emergency could appear and the multiple features and relationships that characterize them make impossible to agree a proper model for the monitoring task.

2.5.4.7 Solution refinement or upgrade

When the accuracy of a certain mechanism is not enough for satisfying the expectations of the system, the logic solution is using the output of this mechanism as an input of another mechanism able to achieve the level of accuracy desired. The integration between both subsystems reaches a co-operative level.

In RIMSAT system, MBR can be used for extending the solution given by the most similar case retrieved applying concrete models to that specific situation. Although the complexity of the whole RIMSAT system is very high and intractable, it can be divided in specific incident environment where a model can be applied to extend the lesson given. The model used is termed local model, as it only deals with that particular case of incident and/or situation, and model-based reasoning is used for refining and extending the solution retrieved by case-based reasoner offering useful information to the fire-fighter. Then, the concrete Model Based Reasoning technique used for the refinement task will depend on what is required to be upgraded. Some valid proposals for this task could be the following:

- A local model for computing the concrete amount of resources needed for facing certain situation.
- A local model for computing the diameter of the area to be evacuated.

- A local model for computing the weight that a certain structure can handle without collapsing given the current state of the building.

2.5.4.8 Revising alternative solutions:

In a concrete situation alternative solutions can appear feasible, and having a system that determines what is the best option basing its decision in some criteria could be a good record. In diagnosis systems this is a technique that has been used. One good example is [Heckerman et al, 1994] where they perform a *value-of-information* analysis in the field of decision analysis. They explicitly calculate the expected value of various observations in terms of their subsequent effect on actions. In their framework, an observation is valuable only if the expected cost of repair given that observation is less than the expected cost of repair in the absence of that observation, and the observation and repair have nonzero cost. After analysing costs and using a probabilistic model they suggest one of two options: revising another possible faulty component or replacing the whole piece with all its elements.

In RIMSAT domain such functionality could result interesting and applicable. It is clearly an MBR task that could be integrated in the Revise phase in the CBR cycle in situations where alternative solutions are retrieved with similar score. One example of such situation could be having one case that suggests the controlled explosion of a truck cargo, and another one that suggests the use of water for extinguish the fire on it. Then, a local model of the truck cargo typical failures can be used requiring some extra information in order to predict if is worth to try to save the truck cargo basing this prediction in a probabilistic model and in a balance of lost and benefits. Once the MBR system has acted, its output could serve to prune the cases that will be presented to the user, or just to change the scores of the presented cases.

2.5.4.9 Factoring cases

As the information gathered by the system, that is, the input from the user, refers to different aspects of the emergency situation, the lesson given should be different in every subcase. For achieving that, the initial information input can be spliten into different cases, each one tracking the relevant features for each actuation area. Hence, RIMSAT will be able to advise the firefighters involved in different aspects of the emergency with the best specific lesson to that arena.

[Branting *et al.*, 1997] uses MBR factoring cases approach in their CARMA system used in a rangeland pest management domain as a tract of rangeland often contains multiple, distinct grasshopper populations composed of species whose consumption characteristics vary greatly. CARMA therefore factors the overall population of a case into subcases according to wintering types using a model of grasshopper developmental stages that estimates the probable hatch and death dates of each grasshopper population given the current development stage, growing season dates for the location, and current date.

Note that for achieving this task a deep knowledge of the domain is needed in order to determine the relevant features that characterize each actuation area.

2.5.4.10 Discovering dependences

In a concrete domain there can be dependences between its components. These dependences in many cases are hidden and finding them can be the final goal of the automatic reasoning process. This discipline can be identified as the fashionable topic of ‘data mining’ and many approaches are being developed for helping in concrete situations.

Nowadays the technology used for data analysis (data mining) is sophisticated enough for dealing with uncertainty, and the most successful methods used are probabilistic, as for example neural networks or Bayesian networks. We would like to highlight one concrete technique for discovering dependences, that is the one presented by Avi Pfeffet et al. in [Pfeffer et al, 1999] where they use a probabilistic Object Oriented Knowledge Representation, an evolved method that endows the classical Bayesian networks with a higher degree of expressivity and with efficient inference methods. In their system, classes are organized in a class hierarchy. A subclass inherits the probability model of its superclass, and it can also override or extend it. The inheritance mechanism facilitates model reuse by allowing the commonalities between different classes to be captured in a common superclass. The method offers also the possibility of allowing the attributes of an object to depend on aggregate properties of a set of related objects via quantifying attributes. They offer also the possibility of modelling structural uncertainty over the cardinality of these aggregations, which is quite interesting, and also reference uncertainty is another kind of structural uncertainty that the method can deal with. They also offer a structure called *dependency graph* that can be used to make sure that all probabilistic influences, including the influences between different objects, are acyclic. But there is always a payoff for extra features. In this case having this extra expressivity dramatically impacts in computation time, and also, big amount of training information is required for constructing the model with enough accuracy.

Trying to apply the advantages of such methods to the RIMSAT arena, we have to consider our requirements and the resources available. The level of uncertainty modelling infrastructure that the most sophisticated data analysis methods offer is indeed enough for dealing with our scenario. However, the main problem in the domain of RIMSAT is the incompleteness that we will have to face on behalf of efficiency in the data retrieval stage of the process. The RIMSAT users will not cover all the needed aspects of the domain as the domain we wish to cover is extremely wide. This lack of information would be reflected in a non reasonable model of dependences, as not taking into account all the elements involved that could justify certain situation will lead to the algorithm of construction of the model to not discover any relation or to discover relations in a non accurate way. Apart from that, for constructing automatically a model, enough training examples must be provided. The wider is the domain to be covered, (and the more hidden dependences it could exist) the more examples we will need. In our case, we will not have this information.

2.5.4.11 Prediction

The integration of predictive skills in order to foreseen the behaviour of the system to avoid restoring the normal situation tardily will diminish both human being and environment looses. Thus, the integration of a new kind of knowledge can overcome this problem. As described in section 2.2.3 Modelling how?, the transition networks is a possible MBR technique used to take into account the temporal component of a system. In [J. Palma-R.Marín, 2002] this idea of modelling the temporal dimension as a network to capture the dynamism of the system is used in medical domains. In [Gimeno et al, 1997] this methology is used in automatic control of industrial processes, more specifically, in wastewater treatment plants processes.

As described in RIMSAT project, the timeline of an incident can be divided into six phases: *pre-incident*, *incident notification*, *in route to incident*, *on arrival*, *during incident* and *post-incident*. The core step is *during the incident* process, where actions in response to the situation are taken leading to an evolved situation that will be a new input for the CBR architecture.

Therefore, the knowledge used to feed the algorithms of the CBR architecture was static, in the sense that no information about the temporal component of the incident was included. This situation differs from the knowledge used by the firefighter who determines the correct actuation taking into account the possible future situations of the incident state. Thus, RIMSAT can incorporate this predictive knowledge as a way of describing the temporal relationships of these different states of the incident to the experts.

2.5.4.12 Information retrieval

Information about a monitored system that is already available to a computer typically does not cost anything to display. However, in time-critical, high-stakes situations, the time required by people to review information, and confusion arising in attempts to process large amounts of data quickly, can lead to costly delays and errors. Miller found that humans cannot consider more than five to nine distinct concepts of information simultaneously. Moreover, people cannot retain and reason simultaneously about more than two concepts in environments filled with distractions. Somehow we need to gather information from a system to reason with, and also gather input from the reality to feed the system (if we are not working in an automatic sensed domain). If the amount of information to be processed or asked to the human user is big, we are under the risk of information overload leading to mistakes, confusion, lost time, etc. This issue has been studied, and some references can be found providing solutions.

In [Horvitz et al., 1995], methods for managing the complexity of information displayed to people responsible for making high-stakes and time-critical decisions are described. These methods are not reusable in our domain as they are centred in filtering the information from the system to the human user in a system that issues many measures from many sensors.

Rather than the approach described above, in RIMSAT domain a great help could be provided in filtering the questions from the system to the user. This problem can be faced from the knowledge management point of view when the domain model is constructed, but some extra help could be provided from an MBR perspective. There exist relations between the variables that are going to be used in RIMSAT for defining the domain. These relations can be considered as dependency/independency in the sense that, for example, knowing that the value of a certain attribute is 'x', then there will be a set of attributes that will not take any value and, therefore, there is no point in querying the user about them. A simple network of dependences with the expressivity of a basic Bayesian Network will be enough for constructing such model, which later on can be used for improving the performance in the information gathering stage using it for filtering the values to fill from the domain as the gathering process advances.

Another helpful use of MBR in RIMSAT during the information retrieval stage could be the coherence checking for the input values. Using a model of correctness, the input can be checked in order to detect mistakes in the data entry.

3 INTEGRATING CBR & MBR IN RIMSAT

The reason of the integration is to take the most of each technology separately, having all their advantages altogether, and trying to compensate their drawbacks. In such a critical scenario as the emergencies domain, is extremely important to keep some of these properties among the others. Reliability and maintainability are essential demands. Both characteristics, in a system like RIMSAT, are ensured by the properties of scalability and learning of CBR combined with a deterministic output of a Model Based Reasoner. In this way, the system should be able either to adapt itself to changes in the existing models (learn) or to introduce changes keeping the integrity of the knowledge base.

As emergencies situations are different one from another, the predictive knowledge is also an important issue as the system has to react to completely new situations. The advantage of MBR over CBR is that the former covers this predictive knowledge thanks to formal methods (logic-like), that is, formal techniques, while the latter uses the experiences already treated. But other possibilities of combining CBR and MBR can be possible at a different level, for example using the formal methods of MBR for helping CBR in some internal task as for example the computation of the similarity between cases.

The different levels of integration between MBR and CBR technologies are explained in the first section of this chapter. Once these levels are explained, the next subsection will make a reflexion about what can be done within RIMSAT context given the state of the art of the research and the technologies available for the implementation of RIMSAT. The third section will define in more technical terms the strategies for integrating both technologies that are definitely feasible...

3.1 INTEGRATION LEVELS OF MBR & CBR

For the purpose of RIMSAT, four possible levels of integration between the CBR and MBR technologies have been identified (see Figure 10):

- The first consists simply of keeping both technologies separate and letting the user organisation (system administrator) choose one or the other approach depending on the circumstances. This toolbox approach should not be rejected because such a user may need only MBR or CBR.

- In the second level of integration, called the co-operative level, the technologies are kept separated but they collaborate. Each uses the results of the other to improve or speed up its own results, or both methods are used simultaneously to reinforce the results. For instance MBR can be used as an input for a case retrieved from the whole process of the CBR engine.
- The third level of integration, called the workbench level, goes a step further. The technologies are separated but a ‘pipeline’ communication, which is used to exchange the results of individual modules of each technology.
- The final level (the seamless level) aims at reusing the best components of each method to build a powerful integrated tool, which avoids the weaknesses of each separated technology and preserves their advantages.

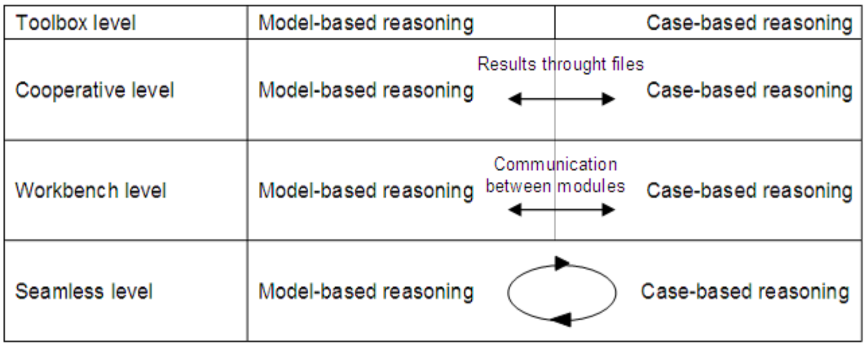


Figure 10. Four Integration Levels in RIMSAT

We have identified several systems[2] that study different perspectives of integration of CBR-MBR techniques. Branting et al. [Branting-Hantings, 1994] study within the CARMA system a model-based matching and adaptation perspective to integrate CBR with MBR for prediction in rangeland ecosystems. They reached the seamless level of integration using a causal model to assist CBR in four ways: factoring cases into subcases, temporal projection, featural adaptation and critical time adjustment.

The ADAPTER system, the work by Portinale et al. [Portinale-Torasso-Ortalda-Giardino] presents another approach to the integration of these technologies within the diagnostic system framework. Such integration is exploited by defining adaptation criteria on solutions retrieved by a case-based reasoner, in order to focus the model-based reasoner in the search for the solution of the current case and avoiding, as much as possible, the computation of the solution from scratch. They reached the co-operative level where MBR is applied for adapting the case retrieved from the case base thanks to CBR.

EPAION, [Karamouzis-Feyock, 1992], is an exemplary case of utilisation of CBR to extend the ability of MBR to deal with external influences that cannot be captured by the model. MBR prognostic predictions are limited by the level of details of the model. There are situations that can not be covered by the model but, however, can be dealt by the experience. Therefore, CBR can complete the model by adding-up new cause-effect links. EPAION reaches the seamless level connecting CBR and MBR components together.

CHEF application, [Birnbaum, Collins et al. 1991], uses MBR to improve the performance of CBR system within the cooking domain. CHEF reaches the workbench level using explicit models of the planning and plan execution processes to generate expectations about the desired performance of the planning system.

[2] For more details, see section 2.3.3

[Bergmann, Pews et al. 1994] investigated the impact of using explanation-based similarity for case-based retrieval and adaptation in the frame of a diagnostic (MOCAS) and a planning (PARIS) application. In this case, the level of integration is the co-operative one.

3.2 AVAILABLE MODELS AND THEIR POSSIBLE USAGE FOR AUTOMATIC REASONING

As a results of the analysis of current user practice undertaken in Wokpackage 2 of the RIMSAT project, we identified that ready to use models do not exist for the purpose of RIMSAT. However, by examining the huge amount of information available, we are able to identify many elements that could be useful for building a model, and also some specific models that captures concretes parts of the domain we are dealing with.

3.2.1 Potential models for RIMSAT

3.2.1.1 Fire propagation model

Stamping geographical information could be useful if a model about propagation of a fire in an open space as, for example, a forest, or a geographical model for planning evacuations in a wide geographical area are going to be constructed. This kind of models can be constructed with certain commercial tools that would be integrated in the system.

3.2.1.2 Combustion model

In general, behavioural prediction models can be constructed for helping in specific problems in our domain. A time model would be useful to predict the behaviour of the system. Using a Transition Network we could capture, for example, the evolution of the combustion of certain materials achieving in this way a proactive behaviour of the fire-fighter that could be very appreciated.

3.2.1.3 Chemical models

A computerised model for simulating the chemical elements' reacting behaviour, for then establishing a categorization that will be the input of the CBR system. Achieving in this way a simplification of the imput for the query in the CBR system.

3.2.1.4 Highway cargo tank behaviour

(From "HAZMAT Bulletin, August/September 1995") - from this interesting article, we can see that the behaviour of a liquid propane tank may be predicted from the physical characteristics of liquid propane and of the tank itself.

3.2.1.5 Forest fires models

Some works have been done on modelling the behaviour of a forest fire. Presumably the behaviour of a forest fire is a bit more foreseeable than other kinds of fires.

3.2.1.6 Resource management

(e.g. water, foam supplies; appliance dimensions; hose lengths) (from the Building Disaster Analysis Group, Her Majesty's Inspectorate, UK)

There are systematic rules, predictive knowledge and information attached to well identified elements of an incident. In the mentioned case: the dimension of the building, the importance of the fire and its nature.

3.2.1.7 Breathing Apparatus models

(effort/air supply/time) Same as above. There exist systematic rules and predictive knowledge and information attached to the use of breathing apparatus.

3.2.1.8 Attribute dependencies models

The knowledge that an expert in our domain has about how the parameters captured are related can be explicited with certain techniques as Object Oriented models, or even with dependencies networks for then using them in helping the computation of the similarity, or deciding how to tear the query for specific user profiles (strategic filters).

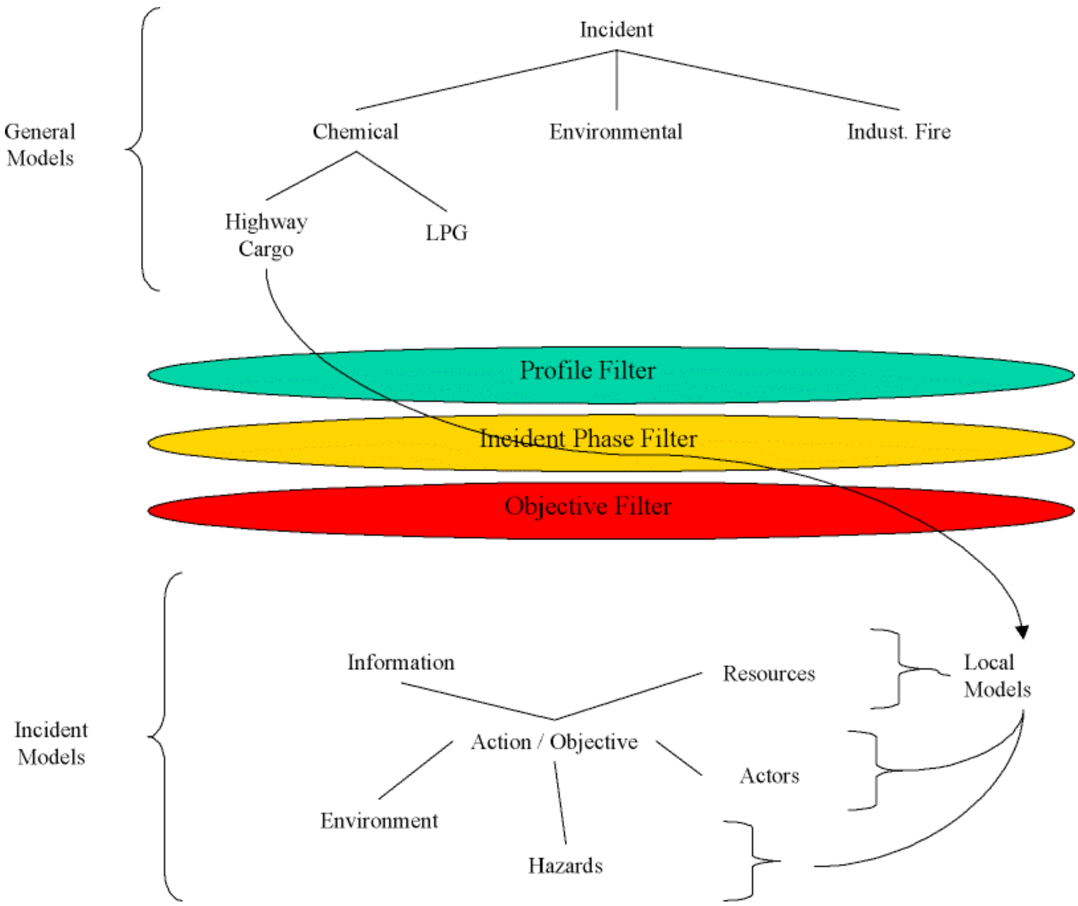
3.2.2 Models Categorisation

Incident type - the fire services already have a typology of incident. (Provisional data requirements for the fire service emergency cover review – Version 2.0)

Starting from the potential elements for modelling, as described in the previous chapter, it is easy to group them into three categories:

- General models: Those which could lead to a general model (i.e. amenable to a model covering any incident or situation);
- Incident model: Those related to an incident (i.e. incident type, kind of situation, …);
- Local model: Those related to an element of an incident scene (a Unit of Experience as per RIMSAT definition).

The positioning of these three models is illustrated below:



Based on the above, we can group the identified models into the three categories:

Potential Generic Models

- Attribute dependencies model

Potential Incident Models

- Forest fires
- Fire propagation models

Potential Local Models

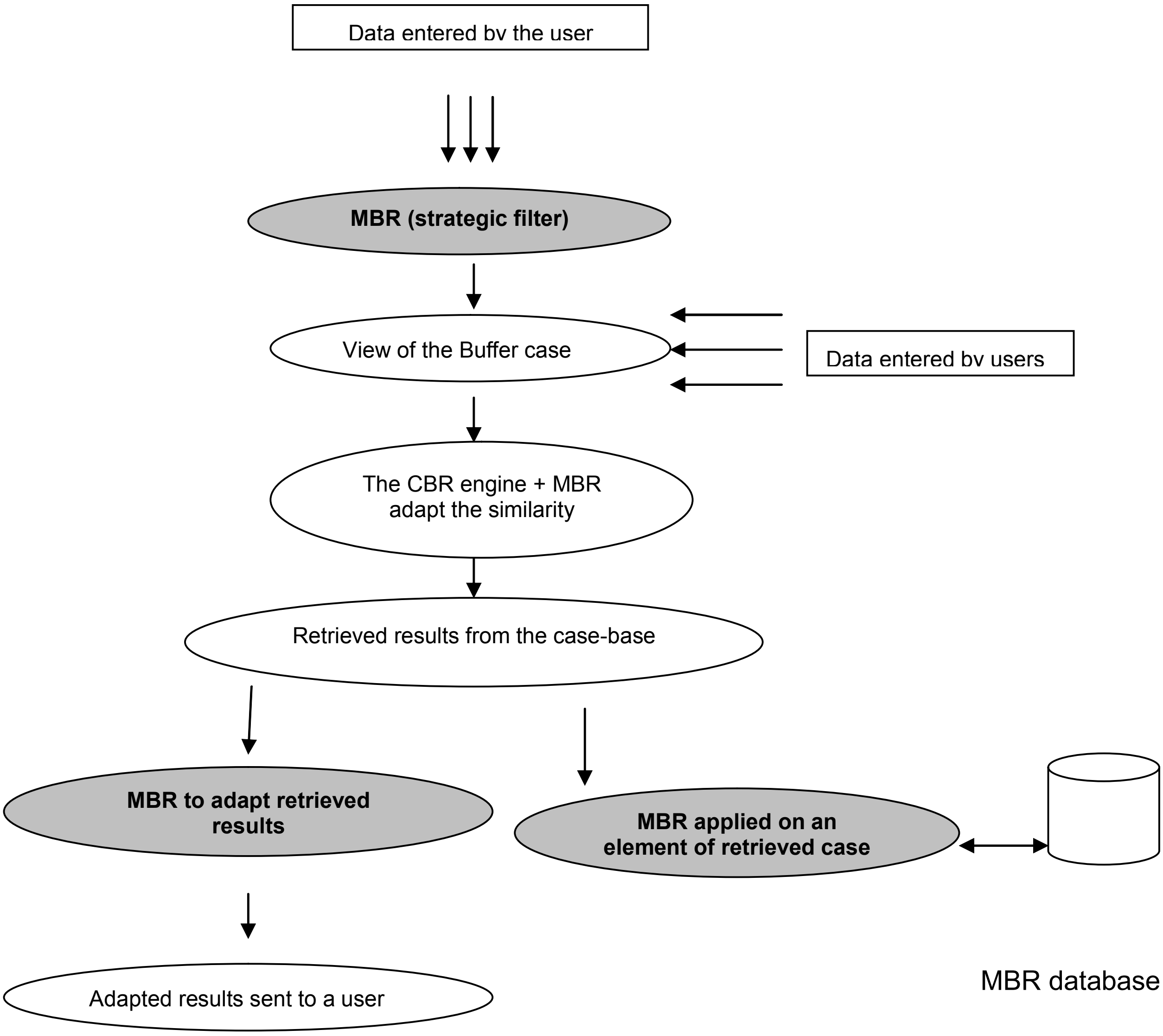
- Highway cargo tank behaviour
- Resource management (e.g. water, foam supplies; appliance dimensions; hose lengths)
- Breathing Apparatus (effort/air supply/time)
- Chemical model

3.3 INTEGRATION AND ARCHITECTURE

3.3.1 Integration strategies for RIMSAT

The MBR and the CBR techniques can be integrated in the RIMSAT system in different ways :

- The system will act in proactive phase, through the strategic filter to get a view of the case query. (More details on 4.2.1)
- In a reactive phase of the system, the MBR will be used to adapt the results retrieved from the case-base to get best recommendations to the current situation.
- To apply an MBR model on an element of the case query. (See next figure)



3.3.2 Integration of CBR & MBR in RIMSAT - an Analysis

3.3.2.1 Overview of requirements

As stated in section 2.5.3 above, RIMSAT will offer 3 types of sessions:

- Incident session will be used during an incident in order to deliver decision support to the user.
- Test and training session will be used offline in order to test the system on specific cases or procedures and to enable the users to get trained on specific aspects of an incident .
- Management session will be used to create and modify the content of RIMSAT (document, cases, models…).

3.3.2.1.1 Incident Session requirements

An incident session will contain all the data captured from the beginning to the end of an incident. All data entered, actions done, documents browsed etc. during an incident will be time stamped and logged. As such, an incident session will be a repository for all information as part of debrief after the end of the incident.

An incident session will be opened and closed by the Incident Commander. When an incident session is opened, data can be entered into the session asynchronously by several users.

During an incident session, the users will receive decision support information depending on their profile, the incident type, the incident phase and the objectives to be fulfilled.

3.3.2.1.2 Test and Training Session requirements

A test / training session will be used to consult offline the content of the RIMSAT system in order to solve a specific problem / question (e.g., by retrieving a 'unit of experience', a specific document, a link to a chemical database etc...). This session will also enable the user to see the system behaviour using 'what-if' scenarios. In this session, no consultation log will be stored.

3.3.2.1.3 Management Session requirements

The management session will enable the contents of the RIMSAT system to be updated and validated - e.g:

- Creating new documents
- Linking to external documents
- Creating new cases
- Validating / modifying existing cases
- Adding/updating existing models.

In addition to the above, RIMSAT will provide:

- Profile management, offering three different profiles of system access and usage;
- Document management - for both internal and external documents;
- Case management - including: debrief support; case creation, modification, suppression; and case validation.
- Model management - enabling the upkeep of models by adding/updating the existing models.
- Data collection - data will be collected during an incident by manual input into the system, asynchronously by several users [NOTE: in test / training mode there will be only one user].
- Ability to handle contradictory information - data input by various sources and users with several profiles will be examined, this will be achieved by checking data consistency and verifying its coherence. Data suspected to be contradictory will be notified to the user who entered the information and the Incident Commander.
- Time dependent data - during an incident the system will take account of time data validity, all data entered will be qualified for a certain time, which will be known a priori.
- Decision support tactics - each time new data is entered by users the system will propose a set of tactics in order to help the user in his decision process. For example: consult a chemical database; compare current situation to previous ones; read a document etc...

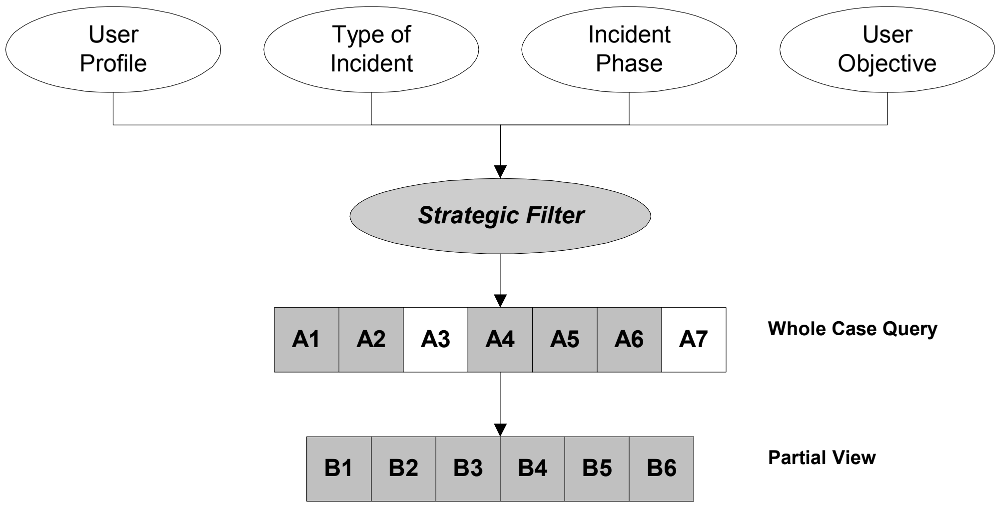
The suitability of CBR, MBR and hybrid CBR/MBR were evaluated against the above sessions' requirements.

3.3.2.1.4 Incident Session CBR/MBR applicability

During an incident, information will be constantly provided to the RIMSAT system remotely and asynchronously by several users. Data will be entered asynchronously by several users. At any time, the users will have the possibility to query the system for a set of current recommendations and / or a set of additional information to be collected. The type of query and the results obtained will depend on the user’s profile, incident category, incident phase and objectives. The questions and recommendations obtained will be linked to a set of information resources, either internal documents or external resources reached by URL hyperlinking over the Web.

CBR has been assessed as suitable for this activity.

Data will be entered depending on the user's **profile**, incident **category**, incident **phase**, and user's **objective** that he/she want to achieve. In other words, the categories will act as filters on the data input. An MBR model can possibly be used as a strategic filter, which will have as entry the following information: the user's profile, the incident category, the incident phase, and the user's objective he/she want to fulfil. The output of this filter will be a view of the case buffer that represents and describes an aspect of the incident - as shown in the following illustration:



With respect to the decisions support tactics mentioned above, the following technology is relevant for specific tactics:

- Consult a chemical database - database-like tactic;
- Compare current situation to previous ones - CBR-like tactic;
- Read a document - document retrieval tactic;

We can justify the applicability of CBR and MBR in Incident Sessions considering what is stated in section **¡Error! No se encuentra el origen de la referencia.** - **¡Error! No se encuentra el origen de la referencia.**. The MBR choice matches the requirements for this part of the system; therefore we can conclude that its usage is justified.

Considering what is also stated in section 3.2.1 - Potential models for RIMSAT we can likewise conclude that some applicable models have been clearly identified, among which:

- Fire propagation model
- Combustion model
- Chemical models
- Highway cargo tank behaviour
- Forest fires models
- Resource management
- Breathing Apparatus models
- Attribute dependencies models

These are local models. The concrete application of such models will mostly intervene in the adaptation of CBR search results.

3.3.2.1.5 Training Session CBR/MBR applicability

When used in training session mode, RIMSAT will essentially operate in the same way as for the incident session mode.

Considerations about applicability for training sessions are the same as for incident sessions.

3.3.2.1.6 Management Session CBR/MBR applicability

With respect to the planned management sessions, CBR and MBR can play a part as follows:

- Creating new cases - CBR
- Validating / modifying existing cases - Hybrid CBR/MBR
- Adding/updating existing models – MBR. For this point, we must be aware that model management will actually mean updating and configuring models to be used in the following situations:
    - Search result adaptation
    - Strategic Filter
    - Input Data coherence check

whereas

- Creating new documents - would use document retrieval technology
- Linking to external documents - would use hyperlinks forwarding to some remote documents (i.e. over the internet).

In the management session we should also contemplate all the administrative tasks such as:

- User management (creation/updating/deletion)
- Role management (creation/updating/deletion) and users-roles association

4 CONCLUSION

After the study of the state of the art of the CBR and MBR integration techniques for emergencies domain, and after the study of the different possibilities of use of both techniques within RIMSAT, we can conclude:

- The concrete MBR techniques we find in the literature are not directly applicable to model RIMSAT scenario and provide an accurate and useful output given the wideness of it. However, they can be fruitful for solving specific problems of the whole domain.
- The local similarity functions that are currently being used in the CBR systems are not optimal for our domain. We are dealing with a domain model whose variables are very co-related, and their inter-dependences are well understood. Such an scenario has suggested us the idea of creating a model with the expressivity of a Bayesian network in order to capture the well known but complex network of dependences between the attributes and using this model for creating a new concept of local similarity weighting function.
- CBR and MBR can be used together at different levels of integration in RIMSAT domain.
- The role of CBR within RIMSAT has been always quite clear. Having a case base for managing the knowledge needed for facing certain situations Conceptually, the kind of tasks that MBR could perform in our environment can be:
    - o Using MBR for manipulating cases in the case base (factoring cases, correcting cases, branching cases...).

- Using MBR for temporal projection (prediction).
- Using MBR for revising alternative solutions
- Using MBR for solution refinement or upgrade.
- Using MBR for similarity assessment.
- Using MBR for information retrieval.

The use of one or another will depend on the available information for implementing the models.

REFERENCES

[Aamodt & Plaza, 1994] A. Aamodt and E. Plaza. Case-based reasoning: fundamental issues, methodological variations and system approaches. *AI Communications* 7(1):39-59, 1994.

[Alonso-Betanzos et al. 2002] A. Alonso-Betanzos, O. Fontela-Romero, B. Guijarro-Berdiñas, E. Hernández-Pereira, J. Canada, E. Jimenez, J. Legido, S. Muñiz, C. Paz-Andrade, M. Paz-Andrade. A Neural Networl Approach for Forestal Fire Risk Estimation. Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Alterman, 1988] R. Alterman. Adaptive planning. *Cognitive Science* 12:393-422, 1988.

[Andriole, 1989] Andriole, S., Handbook of Decision Support Systems, TAB Professional and References Books, Blue Ridge Summit, PA, USA.

[Arcos & Plaza, 1995] J.L. Arcos and E. Plaza. Reflection in NOOS: an Object-centred representation language for knowledge modelling. In *IJCAI Workshop on Reflection and Meta-level architectures and their application in AI (IJCAI'95),* pages 1-10, Montréal, 1995.

[Ashley et al., 1987] K.D. Ashley and E.L.Rissland. Compar and contrast, a test of expertise. Proc 6th AAAI, 1987.

[Ashley, 1990] K.D. Ashley. *Modelling legal argument: reasoning with cases and hypotheticals*. The MIT Press, 1990.

[Avesani *et al*., 2000] P. Avesani, F. Ricci and A. Perini. Interactive Case-based planning for forest fire fighting. Applied Intelligence 13(1):41-58, 2000.

[Babuska et al. 1999] R. Babuska, H.B. Verbruggen, and H. Hellendoorn. Promising fuzzy modeling and control methodologies for industrial applications. In Proc. of European Symposium on Intelligent Techniques ESIT'99, AB-02, Crete, Greece, June 1999.

[Bain, 1986] W. Bain. Case-based reasoning: a computer model of subjective assessment. Ph. D. Dissertation. Dept. of Computer Science. Yale University, 1986.

[Bareiss, 1989] E.R. Bareiss. Exemplar-based knowledge acquisition: a unified approach to concept representation, classification and learning. *Academic Press*, 1989.

[Bergmann, R., G. Pews, et al., 1994] Bergmann, R., G. Pews. Explanation-based similarity for case retrieval and adaptation and its application to diagnosis and planning tasks. Working papers of European Workshop on CBR, Universität Kaiserslautern SEKI: 301-306.

[Beschta et al. ,1993]. A.Beschta, O.Dressler, H.Freitag, M.Montag, P.Struss. A Model-based Approach to Fault Localisation in Power Transmission Networks. *Intelligent Systems Engineering*, 2(1), pp 3-14.

[Birnbaum, Collins, et al., 1991] Birnbaum, L., G. Collins. A Model-Based Approach to the Construction of Adaptative Case-Based Planning Systems. Proceedings of the 1991 Workshop on Case-Based Reasoning, Washington, DC.

[Bonissone et al., 1990] Bonissone P.P. and Dutta S. Integrating case-based and rule-based reasoning: the possibilistic connection. Proc 6th Conf. of Uncertainty in Artificial Intelligence, Cambridge, 1990.

[Bonissone et al., 1999] P. P. Bonissone, Y-T Chen, K. Goebel and P. S. Khedkar. Hybrid Soft Computing Systems: Industrial and Commercial Applications. *Proceedings of the IEE*E, pp 1641--1667, vol. 87, no. 9, September 1999.

[Boverie et al. 2002] S. Boverie, D. Dubois, X. Guerandel, O. Mouzon, H. Prade. Online Diagnosis of Engine Dyno Test Benches: A Possibilistic Approach, Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Brandau *et al*., 1991] R. Brandau, A. Lemmon and C. Lafond. Experience with extended episodes: cases with complex temporal structure. In Proc. of *Workshop on case-based reasoning (DARPA).* Washington D.C., 1991.

[Branting-Hastings, 1994] L. Karl Branting, John D. Hastings. An empirical evaluation of model-based case matching and adaptation, 1994.

[Branting-Hastings, 1995] L. Karl Branting, John D. Hastings. Global and Case-Specific Model-Based Adaptation, 1995.

[Branting *et al.*, 1997] Branting. L.K., Hastings, J.D., and Lockwood, J.A. Integrating cases and models for prediction in biological systems. *AI Applications* 11(1):29-48, 1997.

[Brusoni, Console et al. 1996] V. Brusoni, L. Console. A spectrum of definitions for temporal model based diagnosis: 1-9. 1996.

[Bui, 1999] Bui, T. Building Agent-Based Corporate Information Systems: An application to Telemedicine, European Journal of Operations Research

[Burges 1996] C.J.C. Burges. Simplified support vector decision rules, 1996.

[Carbonell, 1986] J. Carbonell. Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. *Machine Learning* vol. 2, 1986.

[Celler, 1991] F. E. Celler. Continuous System Modelling. New York: Springer-Verlag 1991.

[Cognitive, 1992] Cognitive Systems. *ReMind Developer's Reference Manual*. Boston, 1992.

[Console-Hamscher-deKleer, 1992] W.Hamscher, L.Console, J.deKleer (Eds.). Readings in Model-based Diagnosis. Morgan Kaufmann, San Mateo.

[Console-Torasso, 1991]. L.Console, P.Torasso. A Spectrum of Logical Definitions of Model-based diagnosis. Reprinted in [Console-Hamscher-deKleer, 1992], pp 78-88.

[Console et al., 1991] L.Console, L.Portinale and D.Theseider Dupre. Focusing abductive diagnosis. Proc. 11th Int. Conf. on Expert Systems and Their Applications. Avignon, 1991.

[Cortés, Sànchez-Marrè et al., 2000]. U. Cortes, M. Sanchez-Marre et al. Knowledge Management in Environmental Decision Support Systems. 2nd ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence (BESAI'2000), Berlin, Germany.

[Creecy et al., 1992] R. H. Creecy, B. M. Masand, S. J. Smith and D. L. Waltz, Trading MIPS and memory for knowledge engineering. *Communications of the ACM* 35:48-64, 1992.

[Dague-Deves-Raiman, 1988]. P.Dagues, P.Deves, O.Raiman. Troubleshooting: when Modelling is the Trouble. Published in [Console-Hamscher-deKleer, 1992], pp 218-223.

[Dague et al., 1987]. P.Dague, P.Deves, J.P.Marx, O.Raiman. DEDALE, an expert system for troubleshooting analogue circuits. International Test Conference (IEEE), Washington (USA).

[Dague et. al. 1990]. P.Dague, P.Deves, P.Luciani, P.Taillibert. Analog systems diagnosis. Published in [Console-Hamscher-deKleer, 1992], pp 229-234.

[Dague et al., 1991]. P.Dague, P.Deves, O.Jehl, P.Luciani, P.Taillibert. When oscillators stop oscillating. 12th International Joint Conference on Artificial Intelligence (IJCAI 91), Sydney.

[David et al., 1993] David J.M., Krivine J.P., Simmons R. Second Generation Expert Systems. Springer Verlag, 1993.

[de Kleer, 1986] de Kleer J., An Assumption-based TMS, Artificial Intelligence 28, 1986.

[de Kleer, 1991] de Kleer J. Focusing on probable diagnoses. Proc. AAAI 91, Anaheim, CA, 1991.

[de Kleer-Mackworth-Reiter, 1990]. J.deKleer, A.K.Mackworth, R.Reiter. Characterizing Diagnoses and Systems - A.I.56 (2/3) pp 197-222.

[de Kleer-Williams, 1987] de Kleer J. and Williams B. C., Diagnosing Multiple Faults, Artificial Intelligence 32, 1987.

[Dressler-Struss 1992]. O.Dressler, P.Struss. Back to Defaults: Characterizing and Computing Diagnoses as Coherent Assumption Sets. Proc.of the European Conference on Artificial Intelligence (ECAI), pp 719-723.

[Dressler-Struss 1994]. O.Dressler, P.Struss. Model-based Diagnosis with the Default-based Diagnosis Engine: Effective Control Strategies that Work in Practice. Proc.of the European Conference on Artificial Intelligence (ECAI), pp 677-681.

[Faltings-Struss, 1992]. B.Faltings, P.Struss (Eds.). Recent Advances in Qualitative Physics. MIT Press, Cambridge.

[Fedra-Winkelbauer, 1999] Fedra, K. and L. Winkelbauer. A hybrid expert system, GIS and simulation modelling for environmental and technological risk management. http://www.ess.co.at/docs/papers/aaai99.html

[Fensel, et al., 1997] Fensel, D., E. Motta, et al. The use of ontologies for specifying Tasks and Problem Solving Methods: A Case Study. 10th European Workshop, EKAW, 1997.

[Fox & Das, 2000] John Fox, Subrata Das. Safe and Sound. Artificial Intelligence in Hazardous Applications. AAAI. The MIT Press.

[Garatti et al. 2002] R. Garatti, G. Lamperti, M. Zanella. Diagnosis of Discrete-Event Systems with Model-Based Prospection Knowledge. Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Giere, 1998] Ronals N. Giere. Using Models to Represent Reality. MBR98, Conference Model-Based Reasoning Scientific Discovery Proceedings.

[Gimeno et al, 1997] Gimeno J. M., Béjar J, Sánchez-Marrè M, Cortés U, R-Roda I, Poch M, Lafuente J. Providing WWTP with predictive knowledge based on transiction networks. Proc. Of Int. Conference on Intelligent Information Systems (IIS'96). The Bahamas, 1997.

[Goel, 1991] A. Goel. A model-based approach to case adaptation. In Thirteenth Annual Conference of the Cognitive Science Society, 143-148.

[Goel & Chandrasekaran, 1992] A. Goel and B. Chandrasekaran, Case-based design: a task analysis. In *Artificial Intelligence approaches to Engineering design, vol. 2: Innovative design* (C. Tong and D. Sriram, editors). Academic Press, 1992.

[Greiner, Smith, et al. 1992]. Model-Based Diagnosis. W. Hamscher, L. Console and J. D. Kleer. San Mateo, CA, Morgan Kaufmann**:** 49-53. A correction to the algorithm in Reiter's theory of diagnosis.

[Hammond, 1989] K. Hammond. Case-based planning: viewing planning as a memory task. *Academic Press*, 1989.

[Heckerman et al, 1994] David Heckerman, John S. Breese, Koos Rommelse. Troubleshooting under Uncertainty, 1994.

[Heino et al. 2002] P. Heino, P. Valkokari, K. Rönkkö, E. Kotikunnas, S. Lamberg. Operator Support for Abnormal Situations Using Safety and Reliability Knowledge. Proceedings of the IFAC 2002.

[Heim et al. 2002] B. Heim, S. Gentil, L. Travé-Massuyès, B. Braunschweig. Fault diagnosis of a chemical process using causal uncertain model. Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Hennessy & Hinkle, 1992] D.H. Hennessy and D. Hinkle. Applying case-based reasoning to autoclave loading. *IEEE Expert* 7(5):21-26, 1992.

[Hinrichs, 1992] T.R. Hinrichs. Problem solving in open worlds: a case study in design. *Lawrence Erlbaum*, 1992.

[Horvitz et al., 1995] E. Horvitz, M. Barry. Display of Information for Time-Critical Decision Making. 11th conference of Uncertainty in Artificial Intelligence, Montreal 1995.

[Howe-Cardie, 1997] N. Howe, C. Cardie. Examining locally varying weights for nearest neighbor algorithms. *Proceedings of the Second International Conference on Case-Based Reasoning.* 1997. pp455-466.

[Hunt, 1996] Hunt, J.E., Model Based and Qualitative Reasoning: Industrial Application, 1st International Workshop on Model-Based and Qualitative Reasoning --- Perspectives for Industrial Applications, ECAI '96, Budapest.

[Joh, 1997] D.Y. Joh. CBR in a changing environment. Procc. of *2nd Int. Conf. On Case-based Reasoning (ICCBR'97). LNAI-1266*, pp. 53-62, 1997.

[Jones & Roydhouse, 1995] Jones, E. and Roydhouse, A. Retrieving structured spatial information from large databases: a progress report .Procc. of *IJCAI Workshop on Artificial Intelligence and the Environment*, pp. 49-57, Montréal, 1995.

[Jong et al. 2002] H. de Jong, J. Gouzé, C.Hernandez, M.Page, T.Sari, J. Geiselmann. Dealing with Discontinuities in the Qualitative Simulation of Genetic Regulatory Networks. Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Karamouzis-Feyock, 1992] Karamouzis S. T., Feyock S., An Integration of Case-Based and Model-Based reasoning and its Application to Physical System Faults. Industrial and Engineering applications of Artificial Intelligence and Expert Systems, Ed. F. Belli and F. J. Radermacher. Williamsburg, VA, Springer-Verlag.

[Kass & Leake, 1988] A.M. Kass and D.B. Leake. Case-based reasoning applied to constructing explanations. Proc. of *Workshop on case-based reasoning (DARPA).* Clearwater, Florida. 1988.

[Kersten G. et al., 2000] Kersten G., Mikolajuk Z., Gar-On Yeh A. Decision Support Systems for Sustainable Development; International Development Research Centre and Kluwer Academic Publishers, 2000.

[Khardon & Roth, 1997] R.Khardon, D.Roth. Defaults and Relevance in Model Based Reasoning. *Artificial Intelligence Journal*, special issue on Relevance.

[Kleer & Williams, 1992] W. Hamscher, L.Console and J. D. Kleer. San Mateo, CA, Morgan Kaufman**n:** 100-117. Diagnosing multiple faults**.** Model-Based Diagnosis.

[Klein, 1998] Klein,G. Sources of Power: How People Make Decisions. The MIT Press.

[Klir, 1985] Klir G. J., Architecture of Systems Problem Solving. Plenum Press, New York, 1985.

[Kohavi et al.]R. Kohavi, P. Langley, and Y. Yun. The utility of feature weighting in nearest-neighbour algorithms. http://citeseer.nj.nec.com/kohavi97utility.html

[Kolodner, 1984] J.L. Kolodner. Retrieval and Organization Strategies in Conceptual Memory: a Computed Memory. Laurence Erlbaum, 1984.

[Kolodner, 1985] J.L. Kolodner. Memory for experience. In *The psychology of learning and motivation vol. 19* (G. Bower editor). Academic Press, 1985.

[Kolodner 1987] J.L. Kolodner and R. Kolodner. Using experience in clinical problem solving: Introsuction and framework. IEEE Trans. On Systems, Man and Cybernetics, 1987.

[Kolodner, 1993] J.L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[Kolodner & Simpson, 1989] J.L. Kolodner and R.L. Simpson. The MEDIATOR: analysis of an early case-based problem solver. *Cognitive Science* 13(4):507-549, 1989.

[Koton, 1989] P. Koton. Using experience in learning and problem solving. Ph. D. dissertation. Dept. of Computer Science. MIT, 1989.

[Krovvidy & Wee, 1993] S. Krovvidy and W.G. Wee. Wastewater Treatment Systems from Case-Based Reasoning. *Machine Learning 10*, pp. 341-363, 1993.

[Leake *et al*., 1997] D.B. Leake, A. Kinley and D. Wilson. Case-based similarity assesment: estimating adaptability from experience. Proc. *of American Association of Artificial Intelligence (AAAI-97)*, pp. 674-679, 1997.

[Lee, 1999] M.H. Lee. On models, modelling and the distinctive nature of model-based reasoning. *AI Communications* 12, pp 127-137.

[Lee-Ormsby, 1994]. M.H.Lee, A.R.T.Ormsby. Qualitatively modelling the effects of electrical circuits faults*. A.I.in Engineering* 8, pp 293-300.

[Lenz *et al*. ,1996] M. Lenz, H.-D. Burkhard, P. Pirk, E. Auriol and M. Manago. CBR for diagnosis and decision support. *AI Communications* 9(3):138-146, 1996.

[Liao-Zhang, 1998] T. W. Liao, and Z. Zhang. Similarity measures for retrieval in case-based reasoning systems, *Applied Artificial Intelligence,*12,267-288,1998.

[Long W. J, 1996] William J. Long. Temporal Reasoning for Diagnosis in a Causal Probabilistic Knowledge Based. Artificial Intelligence in Medicine, 8:193-215, 1996.

[Matas, 2000] Matas, J. C. Development, Implementation and Evaluation of an Activated Sludge Supervisory System for the Granollers WWTP. Departament D'Inginyeria Quimica Agrària i Technologies Agroalimentàries, Laboratori d'Enginyeria Quimica i Ambiental. Girona, Universitat de Girona**:** 241.

[Mauss, May et al, 2000] http://monet.aber.ac.uk/summer_school/slides/html/PS-B1e/index_files/frame.html

[Mayer et al. (1) 2002] W. Mayer, Stumpter, D. Wieland, F. Wotawa. Towards an Integrated Debugging Environment. Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Mayer et al. (2) 2002] W. Mayer, Stumpter, D. Wieland, F. Wotawa. Can AI help to improve debugging substantially? Debugging Experiences with Value-Based Models. Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Mohri-Tanaka, 1994] T. Mohri and H. Tanaka. An Optimal Weighting Criterion of Case Indexing for Both Numeric and Symbolic Attributes*, Aha, D. W., editor, Case-Based Reasoning papers from the 1994 workshop,* AAAI Press, Menlo Park, CA.

[Negroponte, 1995] Negroponte, N. Being Digital, New York: A. A. Knopf., 1995

[Núñez et al, 2002] H. Núñez, M. Sànchez-Marrè, U. Cortés, J. Comas, M. Martínez and M. Poch Classifying Environmental System Situations by means of Case-Based Reasoning: a Comparative Study.To appear in Procc of International Environmnetal Modelling & Software Society Conference (iEMSs'2002), Lugano, Switzerland, 2002.[Osborne and Bridge, 1998] H. R. Osborne and D. G. Bridge. A case base similarity framework. Proc. of *4th European Workshop on Case-Based Reasoning (EWCBR'98)*, pp. 309-323, 1998.

[Nuttall, 1990] Nuttal S., NOSTRUM: Constraint Directed Diagnosis, HRCL Technical Report 62 (PhD. Thesis), Human Cognition Research Laboratory, The Open University, England, 1990.

[Osborne and Bridge, 1998] H. R. Osborne and D. G. Bridge. A case base similarity framework. Proc. of 4th European Workshop on Case-Based Reasoning (EWCBR'98), pp. 309-323, 1998.

[Oxman & Voß, 1996] R. Oxman and A. Voß. CBR in design. *AI Communications* 9(3):117-127, 1996.

[Palma-Marín 2002] J. Palma, R. Marín. Modelling Contextual Meta-Knowledge in Temporal Model Based Diagnosis. Proceedings of the 15th European Conference on Artificial Intelligence ECAI 2002.

[Patton et al., 1989] Patton R., Frank P. and Clark R. (eds.) Fault diagnosis in dynamic systems: theory and applications, Prentice-Hall, 1989.

[Perera & Watson, 1996] S. Perera and I. Watson. NIRMANI: an integrated case-based system for strategic design and estimating. *Progress in Case-Based Reasoning. LNAI #1020*. Pp 186-200, 1996.

C. J. Price, Computer based diagnostic systems. Springer-Verlag, 1999.

[Price et al., 1992]. C.J.Price, J.E.Hunt, M.H.Lee, A.R.T. Ormsby. A Model-based Approach to the Automation of Failure Mode Effects Analysis for Design. Proceedings of the IMechE, Part D: the Journal of Automobile Engineering, vol. 206, pp 285-291.

[Pearl J, 1990] Judea Pearl. Readings in Uncertainty Reasoning, Morgan Kaufmann Publishers, Inc., 1990.

[Pearl, J, 2000] Judea Pearl. Causality: Models, Reasoning and Inference. Cambridge University Press, 2000.

[Pews et al., 1993] Pews G. and Wess S. Combining case-based and model-based approaches for diagnostic applications in technical domains. In Proc. EWCBR 93, Kaiserlautern, 1993.

[Pfeffer et al, 1999] A. Pfeffer, D. Koller, B. Milch and K. Takusagawa. SPOOK: A System for Probabilistic Object Oriented Knowledge Representation. Proc. of the 15th annual conference of uncertainty in AI (UAI), Stockholm, Sweden. August 1999.

[Portinale-Torasso-Ortalda-Giardino] Luigi Portinale, Pietro Torasso, Carlo Ortalda, Antonio Giardino. Using Case-Based Reasoning to Focus Model-Based Diagnostic Problem Solving.

[Purvis-Pu, 1995] Purvis L. and P. Pu. Adaptation using constraint satisfaction techniques. Proceedings of the First International Conference on Case-Based Reasoning,.

[Purvis, 1998] Purvis L. Synegry and commonality in Case-Based and constraint-Based Reasoning. AAAI Spring Symposium on multimodal reasoning, Standford University.[Ram, 1993] A. Ram. Indexing, elaboration and refinement: incremental learning of explanatory cases. *Machine Learning* 10(3):201-248, 1993.

[Ram, 1993] A. Ram. Indexing, elaboration and refinement: incremental learning of explanatory cases. *Machine Learning* 10(3):201-248, 1993.

[Ram & Hunter, 1992] A. Ram and L. Hunter. Goals for learning and understanding. *Applied Intelligence* 2(1):47-73, 1992.

[Redmond, 1992] M.A. Redmond. Learning by observing and understanding expert problem solving. Georgia Institute of Technology. College of Computing. Technical report GIT-CC-92/43, 1992.

[Reiter, 1987] R.Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*. 32 . pp 57-95.

[Reiter, 1992]. W. Hamscher, L. Console and J. D. Kleer. San Mateo, CA, Morgan Kaufman**n:** 29-48. A theory of diagnosis from first principles. Model-Based Diagnosis.

[Riesbeck & Schank, 1989] C.K. Riesbeck and R.C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates Publishers, 1989.

[Rissland et al., 1989] Rissland E.L. and Skalak D.B. Combining case-based and rule-based reasoning: a heuristic approach. In Proc. 11th IJCAI, Detroit, 1989.

[Sacerdoti, 1977] E.D. Sacerdoti. *A structure for plans and behavior*. North-Holland, 1977.

[Sànchez-Marrè *et al*., 1997a] M. Sànchez-Marrè, U. Cortés, I. R.-Roda, M. Poch, and J. Lafuente. Learning and Adaptation in WWTP through Case-Based Reasoning. Special issue on Machine Learning of *Microcomputers in Civil Engineering* 12(4):251-266. July, 1997.

[Sànchez-Marrè *et al*., 1997b] M. Sànchez-Marrè, U. Cortés, J. Béjar, J. De Gràcia, J. Lafuente and M. Poch. Concept Formation in WWTP by means of Classification Techniques: a Compared Study. *Applied Intelligence* 7(2):147-166. April, 1997.

[Sànchez-Marrè *et al*., 1998] M. Sànchez-Marrè, U. Cortés, I. R.-Roda, and M. Poch. *L'Eixample* distance: a new similarity measure for case retrieval. Procc. *of 1st Catalan Conference on Artificial Intelligence (CCIA'98),* pp. 246-253.Tarragona, Catalonia, EU, 1998.

[Sànchez-Marrè, Cortès, et al. 1998] M. Sànchez-Marrè, U. Cortés. Integration of a ruled-based expert system, a case-based reasoner and an ontological knowledge-base in the wastewater domain. ECAI 98 W7 Binding Environmnenttal Sciences and AI.

[Sànchez-Marrè, Cortès, et al. 1999]. M. Sànchez-Marrè, U. Cortés**.** Sustainable case learning for continuous domains. Environmental Modelling & Software 14: 349-357.

[Sànchez-Marrè *et al*., 1999] M. Sànchez-Marrè, U. Cortés, I. R.-Roda, and M. Poch. Sustainable case learning for continuous domains. *Environmental Modelling & Software* 14(5):349-358, 1999.

[Schank & Abelson, 1977] R. Schank and R. Abelson. *Scripts, plans, goals and understanding*. Lawrence Erlbaum, 1977.

[Schmidt 1996] M. Schmidt. Identifying speakers with support vectors networks. In *Proceedings of Interface '96*, Sydney, July 1996.

[Scholkopf et al. 1995] B. Schölkopf, C.Burges, and V.Vapnik. Extracting support data for a given task. In U.M. Fayyad and R.Uthurusamy, editors, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, Menlo Park, CA, 1995. AAAI Press.

[Shinn, 1988] H.S. Shinn. Abstractional analogy: a model of analogical reasoning. Proc. of *Workshop on case-based reasoning (DARPA).* Clearwater, Florida. 1988.

[Stanfill-Waltz, 1986] C. Stanfill, D. Waltz. Toward Memory-Based Reasoning, *Communications of the ACM.* 1986.

[Simmons, 1992] Reid Simmons, *Concurrent Planning and Execution for Autonomous Robots, IEEE Control Systems*, 12:1, pp. 46-50, February 1992.

[Sprague, 1983] Sprague, R., A framework for the development of Decision Support Systems. Decision Support Systems: A Data Based, Model-Oriented, User-Developed Discipline, Petrochelli, Princetown, NJ, USA.

[Stanford-URL] http://robotics.stanford.edu/~koller/#research

[Struss 1988]. P.Struss. Mathematical Aspects of Qualitative Reasoning. *Intern.J.of A.I. in Engineering* 3(3), pp 156-169.

[Struss, 1989]. P.Struss - Diagnosis as a Process. Workshop Paris 1989. Published in:[Console-Hamscher-deKleer, 1992], pp. 408-418.

[Struss, 1992]. P.Struss. What's in SD? Towards a Theory of Modelling for Diagnosis. Published in [Console-Hamscher-deKleer, 1992], pp 419-449.

[Struss, 1993]. P.Struss. (Multiple-) Model-based Diagnosis Applied to Power Networks. In Proceedings of the 2nd European Control Conference (ECC).

[Struss-Dressler, 1989]. P.Struss, O.Dressler. Physical Negation: Integrating Fault Models into the General Diagnostic Engine. Reprinted in: [Console-Hamscher-deKleer, 1992], pp 153-158.

[Sugeno et al. 1993] M. Sugeno and T. Yasukawa. A Fuzzy Logic Based Approach to Qualitative Modeling. IEEE Transactions on Fuzzy Systems, 1(1): 7-31 (1993).

[Surma & Brauschweig, 1996] J. Surma and B. Brauschweig. Case-Based Retrieval in Process Engineering: Supporting Design by Reusing Flowsheets. Engineering Applications of Artificial Intelligence. 9(4): 385-391, 1996.

[Sussman, 1975] G.J. Sussman. *A computer model of skill acquisition*. American Elsevier, 1975.

[Sycara, 1987] K. Sycara. Finding creative solutions in adversarial impasses. Proc. *of 9th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum, 1987.

[Takagi et al. 1985] T. Takagi and M. Sugeno. Fuzzy Identification of Systems and its Applications to Modeling and Control. *IEEE Transactions on Systems, Man, and Cybernetic*s, **SMC-15** (1), pp. 116-132 1985.

[Tatar, 1994]. M.Tatar. Combining the Lazy Label Evaluation with Focusing Techniques in an ATMS. Proceedings of ECAI 94, Amsterdam.

[Tatar-Iwanowski, 1994]. M.Tatar, S.Iwanowski. Aspects of Efficient Focusing. Proceedings 5th Workshop on Principles of Diagnosis (DX 94), New Paltz (USA).

[Torasso et al., 1992] Torasso P., Portinale L., Console L. And Casassa Mont M. Approximate reasoning in a system combining prototypical knowledge with case-based reasoning. Fuzzy Logic for the Management of Uncertainty. John Wiley&Sons, 1992.

[Vapnik 1995] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[Veloso & Carbonell, 1993] M.M. Veloso and J.G. Carbonell. Derivational Analogy in PRODIGY: automating case acquisition, storage and utilization. *Machine Learning* 10(3):249-278, 1993.

[Veloso, 1994] M.M. Veloso. Planning and learning by Analogical Reasoning. Lecture Notes in Artificial Intelligence 886, Springer-Verlag, 1994

[Veloso *et al*., 1996] M. Veloso, H. Muñoz-Avila and R. Bergmann. Case-based planning: selected methods and systems. *AI Communications* 9(3):128-137, 1996.

[Warren *et al*., 1998] T. Warren Liao, Z. Zhang and C.R. Mount. Similarity measures for retrieval in case-based reasoning systems. *Applied Artificial Intelligence* 12:267-288, 1998.

[Watson, 1996] I. Watson. An introduction to Case-based reasoning. *Progress in Case-Based Reasoning. LNAI #1020*. Pp 3-16, 1996.

[Weld-de Kleer, 1990]. D.S.Weld, J.deKleer (Eds.). Readings in Qualitative Reasoning about Physical Systems. Morgan Kaufmann, San Mateo.

[Wettschereck et al, 1997] D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, Special Issue on lazy learning Algorithms, 1997.

[Wierzbicki et al, 2000] Andrzej Wierzbicki, Marek Makowski, Jaap Wessels. Model-Based Decision Support Methology with Environmental Applications. IIASA. Kluwer Academic Publishers, 2000.

[Wilson-Martínez, 1997] D. R. Wilson and T. R. Martínez. Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research, 6, 1-34, 1997.

[Zdráhal, 1992] Zdenek Zdráhal. Existing Model-based Techniques Analisys. HCRL Technical Report (Vital series) TR86 February 1992.

GLOSSARY

Term	Definition
Active	The state of a unit of experience when it is validated and authorised for current use by RIMSAT Operators.
ADR	European system which applies to tank vehicles and tank containers carrying certain substances.
archive	Moving the unit of experience out of the live system and into the library of inactive units of experience, so that they will not be used by RIMSAT Operators.
Attribute	a descriptor which provides insight into the properties of something. In RIMSAT, attributes have a range of associated symbolic or numeric values that can be used to define the properties of the element being described.
Attribute-Value Pair.	A unit of experience can be represented as a flat set of attribute-value pairs where each pair encodes the value of a certain attribute for that unit of experience.
Author (RIMSAT Author)	A person trained to create units of experience.
Awareness.	The success of the project depends on all prospective users being made aware both of its implications and of the possibility that “there is something in it for them,” thus generating motivation.
Basic unit of experience	A unit of experience explaining the application of standard operating procedure in routine circumstances.
BDAG	Building Disaster Assessment Group. A group associated with Her Majesty’s Inspectorate, charged with creating tools to assess the resource needs for specific interventions.
CACFOA	Chief & Assistant Chief Fire Officers' Association. An association of fire officers in the UK.
CACFOA - MDWG	CACFOA Mobile Data Working Group. A subgroup of CACFOA tasked with creating a methodology for procuring mobile data systems.
Case	Unit of experience in RIMSAT.
Case Base.	This is the collection of all available cases (units of experience).
Case Based Reasoning Cycle.	The case based reasoning cycle describes the basic steps involved in case-based problem solving: *retrieve*, *reuse*, *revise*, and *retain*.
Case Buffer.	This is a temporary location where current cases are stored pending a decision about whether or not to edit them into the case base.
Case comparison	Knowledge elicitation technique in which the expert explains the important differences between one incident and another.
CBR	Case Based Reasoning. Case Based Reasoning (CBR) tries to model the acting by experience. It maintains a memory of experiences (*case base*) and solves new problems by *retrieving* similar cases from the cases.
Characteristic	a differentiating feature of an element which helps to tell it apart from others.

ChemData	System used to retrieve details of hazardous materials. Owned by the UK National Chemical Emergency Centre.
Comparison of alternatives	Knowledge elicitation technique in which the expert explains why specific alternative methods were not used.
Context Details	The Context Details section of the unit of experience outlines the key elements of the situation from which the unit of experience was derived
Corporate Memory.	When corporate knowledge becomes easily retrievable for decision support, one speaks of a “corporate memory.”
Cues	Cues in the Incident Analysis Process are all of the elements of an incident which have a special significance to the expert. A cue suggests that there is something specific taking place, that a specific action should be taken, or that a certain risk is present. A key indicator.
Data model	a universe of discourse organised in such a way as to allow easy interpretation by automated systems
Daughter element	An inheriting element in a taxonomy; daughter elements inherit all properties of parent elements
Decision Tree.	This is one form of the output of an “induction engine.” It displays successive partitions of a case base into subsets differentiated from each other by the values of parameters. The first parameter selected for the subdivision is the one generating the greatest information gain as regards factors leading to the target outcome. The subdivision process is then repeated for all other parameters within each branch. See also *fault tree*.
Decomposition	A breakdown of elements into their constituent parts
Degree of belief	the extent to which the RIMSAT Author and RIMSAT Manager believe that the lesson is appropriate for the circumstances.
Deliver	Bringing the unit of experience to the RIMSAT Operator, preferably at the right time and in the right situation.
Dependency	When characteristics of one part of the domain model (value or attribute) are determined by the values specified in other parts of the domain model.
Deployment.	When a system has been developed to the satisfaction of the users, it is usual to generate a “runtime version” of the software, in which all changes made to date are embedded robustly, and system users can use it without help from the developers. The system is then said to be “deployed.”
Description Analysis	See *List Analysis.*
Domain Model	Formal, structured expression of the Universe of Discourse according to the objective of the user. In the *structural CBR approach*, the domain model is a set of attributes, with either defined sets of symbolic values or defined ranges of numerical values sufficient to characterise each unit of knowledge in the knowledge domain. Each case is represented using the attributes, each of which is given one of the allowed values.
Dynamic weighting	The fluctuation of weights of attribute-value pairs based on the specification of other attribute-value pairs
E&D	Exploitation & Dissemination Committee
Effectiveness	The ability to achieve a given goal.

Event	Occurrence producing knowledge about an incident or incident command
Experimental unit of experience	A unit of experience with the lowest "degree of belief". A unit of experience giving advice which was useful in the given context, but which should be followed with extreme caution.
Expert unit of experience	A unit of experience which gives advice on how to solve very difficult or unusual situations by adapting standard operation procedures or through non-standard procedures.
Explanation	The Explanation is a narrative description of the situation, perhaps including such elements as risks, constraints, incident history, the decision that was made, and the consequences. This is where the person logging the unit of experience tells the story of the situation from which the lesson was derived
Explicit Knowledge	Knowledge that is written or certainly retrievable
Fat Client.	When the CBR system is working in client-server mode, the master-version of the system resides on the server. However, for the system to work at an acceptable level of performance, a client with lots of computation capabilities might be necessary. This is called a "fat client," since it could require significant time to download to the client machine.
Flat Model	A single-level model in which all attributes apply directly to the entity being modelled rather than to subordinate parts.
Free-text value	The values of the attribute are free text
Generic model	Model which could cover any incident or situation in the domain.
GIS	Geographical Information System.
Grade-X	A toolset for developing diagnostic systems by integrating a variety of techniques. Owned by GenRad.
GUI (Graphical User Interface).	The graphical user interface is usually customised interactively with the user to meet the user's needs. The information should be presented without overwhelming the user.
Hazard	The inherent potential of something to cause harm.
Hazardous Material	Any materials exposed on an emergency scene that are hazardous by being poisonous, flammable, explosive, carcinogenic, or environmental pollutants. This is also known as hazmat in the emergency services.
HAZCHEM Code	See Emergency Action Code.
HAZMEAT	Official list of hazardous materials used by Fire services and other emergency personnel
Hierarchical Model.	Hierarchical models breakdown a complex system into a hierarchy of sub models, each of which has a greater degree of specificity over the level above.
IAP	See *Incident Analysis Process*
IEEE	Institute of Electrical and Electronics Engineers. International standards organisation.
Inactive	The state of a unit of experience when it is not authorised for current use by RIMSAT Operators.
Incident Analysis Process (IAP)	The process through which units of experience are created from events. A three step process of Telling the Story, Identifying the

	Lessons, and Mapping the Context to Lessons.
Incident model	Model related to particular types of incidents (e.g. dynamic weighting, forest fires)
Incident Reference Number	The code number ascribed to the incident by the organisation.
Induction.	Induction is the generation of rules or decision trees for achieving a desired outcome. The rules or decision trees are generated automatically from the analysis of cases (units of experience) in a case base. This induction process abstracts from the experience of many expert decisions stored as cases(units of experience) in the case base.
Initial Domain.	This is a subset of the total target domain that is used in initial trials of the CBR approach. It should be selected so that positively perceived results are obtained rapidly, thus creating *awareness*.
Integer value	The values of the attribute are whole numbers
Integration.	Integration means bringing together the various parts of the application: The search engine, GUIs, case base, related database, and so on.
Interface background	An area of the user interface that is only visible through a connecting link or other connector from the interface foreground.
Interface foreground	The prominent, visible part of the user interface.
Interface language	A language for a particular interface. The interface language may change according to country or even brigade, but each term in the interface language corresponds to an unchanging term in the domain model.
Intermediate unit of experience	A unit of experience explaining how to adapt standard operating procedures to non-routine situations.
Interruption Analysis	Knowledge elicitation technique, often used with protocol analysis, in which the knowledge engineer interrupts the expert at critical points in the task to ask questions about why they performed a particular action" (Burge 10).
Kaidara Advisor	Development framework for building decision support systems based on case-based reasoning and induction technologies. Owned by Kaidara.
KE	Knowledge Engineering
KM	Knowledge Management
Knowledge	Exploitable comprehension of something, reinforced by experience.
Knowledge Base.	Knowledge base is a generic word for an assembly of chunks of formally represented, distilled knowledge, some or all of which may be in the form of *cases*.
Knowledge Container.	Knowledge container model, introduced by Michael Richter (see Richter 1998), describes the knowledge that a CBR system uses. The containers are the *vocabulary*, i.e., the *domain model*, the *case base*, the *similarity measure*, and the *adaptation knowledge*. In principle, each container could be used to represent most of the knowledge, but for efficient application development it must be carefully decided which knowledge to put into which container.
Knowledge Domain	The scope of knowledge to be treated in a project

Knowledge Life-Cycle	The cycle of creating, modifying, utilising, communicating, and capitalising on knowledge
Lesson	a recommendation for a certain situation
Level of lesson	The indication to the user of the complexity, sophistication, and "degree of belief" of the unit of experience.
List Analysis	Creation and analysis of lists as a way of gaining insight into something.
Local model	Model related to an element of an incident (e.g. breathing apparatus, hose length, etc.)
Manager (RIMSAT Manager)	She/he has all rights over the system, including the right to update the domain model. Updating should be done occasionally as new experiences arise. She/he is responsible for maintaining the *effectiveness* and *relevance* of the RIMSAT system and all of the units of experience within it
MBR	Model Based Reasoning
Methodology.	A methodology is a collection of methods and guidelines that enables a person to work effectively and efficiently in the domain for which the methodology has been developed.
Nearest Neighbour Retrieval.	This is a search approach that selects experience based on some geometrical distance computed in the attribute space. The search engine evaluates the n-dimensional "distance" between the query and all cases in the case base, taking into account the weights. The results are presented in order of n-dimensional "proximity."
Network (RIMSAT Network)	The case base shared among all organisations using RIMSAT, to which all organisations publish there units of experience (in addition to the local RIMSAT system), and from which organisations may pull new units of experience.
Network Manager (RIMSAT Network Manager)	The person in charge of maintaining the RIMSAT Network. He/she authorises changes to the RIMSAT Network case base and domain model.
Objective	The goal at a given time in a given situation.
Object-Oriented Model (OOM)	Model in which entities (objects) are used as symbols of their constituent parts.
Observation	Knowledge elicitation technique in which the expert performs or pretends to perform a task and the knowledge engineer observes, making notes and asking follow-up questions at the end.
Ontology	a model providing standard definitions and relations for a set of elements in a specific domain, often allowing disparate entities to communicate with a common understanding of the domain.
Operator (RIMSAT Operator)	Someone trained to use RIMSAT as a decision support tool for incident command.
Ordered symbolic value	The values of the attribute are nominal symbols in a given sequence
Organisational Process.	Organisational processes cover those parts of the business process that need to be changed in order to make best use of a new software system. They address those parts of the user organisation's business process in which the software system will be embedded. New processes have to be introduced into an existing business process, such as the training of end-users or the technical maintenance of the

	system. Existing processes may need to be changed or reorganised in order to make best use of the new software system.
PAGE	Perceptions, Actions, Goals, Environment. Ontological model used as a way of dividing up data models, universes of discourse etc. into PAGE elements
Parent element	An element which is sufficiently general to head a category of elements (see daughter elements), all of which contain all of the properties of the parent element.
PB	Project Board
Point of view	Someone's perspective at a given time in a given situation.
Precision.	This is the proportion of retrieved cases that turn out to be relevant to a user who needs specific knowledge.
Preferred recipient	the person to whom the unit of experience should ideally be directed
Primary sources	Sources of information or knowledge (such as people) which took content directly from the incident in question.
ProCarta	A commercial tool used by WP2 for representing and characterising processes, activities, and tasks.
Proposition Number	The reference number for the unit of experience before it is approved and entered officially into the RIMSAT system.
Protocol analysis	Knowledge elicitation technique in which the expert performs a task to be analysed while "thinking aloud" (Burge 9).
PSS	Public Safety Services - acronym used throughout the project for all national emergency services such as fire and civil defence authorities.
Publish	Putting the unit of experience into the live RIMSAT system for use by RIMSAT Operators
PVSS	Product Vision Statement and Scope. The rough functionality specification of the RIMSAT tool resulting from the reconciliation of the technical capabilities with the user wish-list.
Quantitative review	A review of a unit of experience based on the number of times it was consulted.
Query path	The process that must be followed to specify the desired search parameters.
QuickPlace	RIMSAT Web-based project work space accessible only by the project partners, EC and Project Reviewers.
RBR	Rule Based Reasoning
Real value	The values of the attribute are real numbers
Recall.	This is the proportion of relevant units of experience from the case base (in the context of the user's current knowledge need) that where retrieved by the retrieval engine.
Reference Number	The unique RIMSAT code number assigned to the unit of experience, once it has been approved.
Relevance	The degree to which something is useful and up-to-date.
Retain.	The retain phase is the fourth step in the *CBR cycle*. Retain means storing new experience in the case base.
Retrieval Engine.	This is a software component that performs the retrieval, i.e., it selects a unit of experience from the case base.

Retrieval.	The retrieval phase is the first step in the *CBR cycle*. Retrieval means selecting a relevant case from the case base. There are different techniques for retrieval, like traversing an induction tree or nearest neighbour retrieval.
Reuse.	The reuse phase is the second step in the *CBR cycle*. Reuse is the application of a unit of experience in the case base to the current problem.
Review	Evaluating existing units of experience according to efficiency and relevance.
Revise.	The revise phase is the third step in the *CBR cycle*. During revision the proposed solution unit of experience is applied and evaluated in the business environment. If necessary, the proposed solution can be improved.
Risk	The danger posed by a hazard in a particular set of circumstances. (Klein, Roger "Command and Hazardous Materials")
RPD	Recognition-Primed Decision Making – a concept defined by Gary Klein which suggests that experienced, safety-critical decision-makers tend to match present situations with past experiences and take the first viable course of action rather than compare various possible courses of action.
SADT	Structured Analysis Design Technique. SADT modelling is a standard way of visualising processes. The basic SADT model can be summed up in one sentence: "Under certain restrictions, inputs are transformed into outputs by a mechanism,"
Safety-critical	Condition in which people's safety is at stake; people's safety depends on the effective resolution of the problem.
Secondary sources	Sources that were not directly in contact with the incident or event in question.
Similarity Measure.	A similarity measure is a computational function that computes the similarity between a unit of experience and a query. The similarity measure contains expert knowledge that evaluates whether a unit of experience contains information that is reusable in the current context defined by the query.
Similarity rating	A percentage value indicating the percent of values in the retrieved unit of experience which match the values specified in the unit of experience query.
Standard reference of discourse	A document which is assumed to be understood by the target population.
State	The condition of an element, usually variable according to time or situation.
Strategic filter	A mechanism that will prioritise information coming out of the RIMSAT system according to specified criteria, e.g. Incident Type, Incident Phase, and Profile.
Structural CBR Approach.	This is a CBR approach that relies on cases that are described with a set of predefined attributes. These attributes are described in a *domain model*.
Symbolic value	The values of the attribute are non-ordered nominal symbols
Tacit Knowledge	Knowledge that is not written, usually held in people's memory

TADMUS	Tactical Decision Making Under Stress – a project researching decision support tools, led by the US Navy
Taxonomic value	The symbolic values of the attribute are linked by a hierarchy structure
Taxonomy	A classification of elements according to groups with similar properties.
TC	Technical Committee
Teachback	Knowledge elicitation technique in which “the knowledge engineer attempts to teach the information back to the expert, who then provides corrections and fills in gaps” (Burge 13).
Textual CBR Approach.	In this CBR approach, cases are represented in free-text form. Keyword matching techniques are used for retrieval. There is no *domain model*.
Time-critical	The condition in which time is a very important limiting factor.
Time-Dependent data	1. Data which has a positive value for dynamism. Data that is subject to change periodically. 2. Data affected by a temporal component in a way that is not static over time.
Time-Stamping	
Top-down Object-Oriented Model	An object-oriented model in which universal attributes are specified at the higher levels in a taxonomy model (and thus apply to all elements below). See *Object-Oriented Model*.
Trainer (RIMSAT Trainer)	The person responsible for training personnel to be certified RIMSAT Authors, Operators, and Managers
UN Number	The number given to classified hazardous materials by the United Nations.
Unit of experience	A lesson drawn from an event, presented with an explanation and the key context information from that event, so that a recipient can understand when and how to apply the lesson. RIMSAT’s version of a case.
Universe of Discourse	A subset of the real world which contains all relevant concepts.
user	The end-user. The person who will be manipulating and/or benefiting from the RIMSAT apparatus.
User wish-list	The list of functions desired by the user-group.
User-group	a cross-section of the population which will use the tool. In the case of RIMSAT, incident commanders and other commanding officers at HAZMAT Transportation and Industrial Fire incidents.
Value	The number or symbol selected to represent a characteristic of an incident.
Weight	A positive integer value representing the relative importance of one attribute-value pair versus another.