

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Using CBR as Design Methodology for Developing Adaptable Decision Support Systems

Hugo López-Fernández, Florentino Fdez-Riverola,  
Miguel Reboiro-Jato, Daniel Glez-Peña and José R. Méndez  
*University of Vigo  
Spain*

## 1. Introduction

Although knowledge-based systems (KBS), and more generally decision support systems (DSS), represent one of the commercial successes resulting from artificial intelligence (AI) research, their developers have repeatedly encountered several problems covering their whole life cycle (Watson, 1997). In this context, knowledge elicitation as well as system implementation, adaptation and maintenance are non trivial issues to be dealt with. With the aim of overcoming these problems, Schank (1982) proposed a revolutionary approach called case-based reasoning (CBR), which is in effect, a model of human reasoning. The idea underlying CBR is that people frequently rely on previous problem-solving experiences when facing up new problems. This assertion may be verified in many day to day problem-solving situations by simple observation or by psychological experimentation (Klein & Whitaker, 1988). Since the ideas underlying case-based reasoning were first established, CBR systems have been found to be successful in a wide range of application areas (Kolodner, 1993; Watson, 1997; Pal et al. 2000). Motivated by the outstanding achievements obtained, some relevant conferences (i.e. ECCBR<sup>1</sup> and ICCBR<sup>2</sup>) and international journals (e.g.: International Journal Transactions on Case-Based Reasoning) have successfully grown up in the field.

In this chapter we present key aspects related with the application of CBR methodology to the construction of adaptable decision support systems. The rest of the chapter is organized as follows: Section 2 introduces an overview about CBR life cycle and combination strategies for constructing hybrid AI systems. Section 3 introduces and covers the main characteristics of four successful decision support systems developed following CBR principles. Finally, Section 4 summarizes the main conclusions and presents the fundamental advantages of adopting this methodology.

## 2. CBR life cycle and combination strategies

A case-based reasoning system solves new problems by adapting solutions that were used to solve previous problems (Riesbeck & Schank, 1989). The case base holds a number of

---

<sup>1</sup> <http://www.eccbr.org/>

<sup>2</sup> <http://www.iccbr.org/>

cases, each of which represents a problem together with its corresponding solution. Once a new problem arises, a possible solution to it is obtained by retrieving similar cases from the case base and studying their recorded solutions. A CBR system is dynamic in the sense that, in operation, cases representing new problems together with their solutions are added to the case base, redundant cases are eliminated and others are created by combining existing cases.

Every time a CBR system copes with a new problem situation, retrieves previously stored cases together with their solutions, matches them against the new problem context, adapts previous outcomes to provide an answer to the new problem and stores the new solution by adding a novel case in the case base. All of these actions are self-contained and can be represented by a cyclic sequence of processes (see Figure 1) in which human interaction may be needed. A typical CBR system is composed of four sequential steps which are called into action each time a new problem is to be solved (Watson, 1997; Kolodner, 1993; Aamodt & Plaza, 1994): (i) *retrieve* the most relevant case(s) (ii) *reuse* the case(s) in an attempt to resolve the problem, (iii) *revise* the proposed solution if necessary and (iv) *retain* the new solution as a part of a new case.

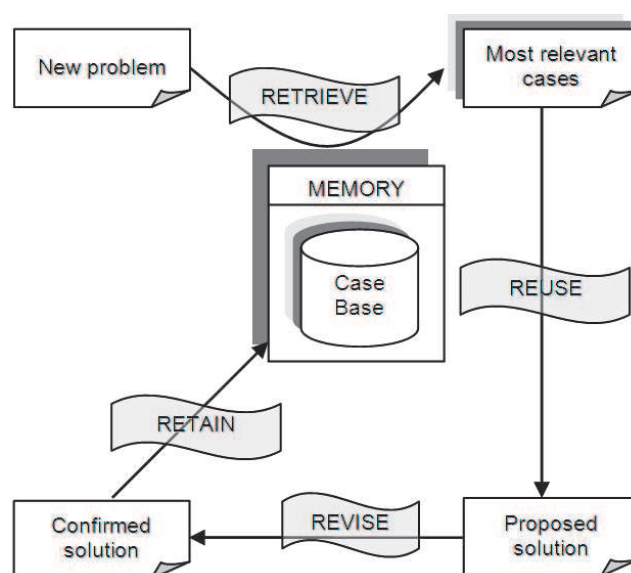


Fig. 1. Typical CBR life cycle comprising four stages

Each of the steps comprising the CBR life cycle defined in Figure 1 requires a model or method in order to accurately perform its mission (see Figure 2). The purpose of the retrieval step is to search the case base and select one or more previous cases that most closely match the new problem situation, together with their solutions. The selected cases are reused to generate a solution appropriate to the current problem situation. This solution is revised if necessary and finally, the new case (i.e. the problem description together with the obtained solution) is stored in the case base. In the CBR cycle there is normally some human interaction. Whilst case retrieval and reuse may be automated, case revision and retention are often undertaken by human experts. This is a current weakness of CBR systems and one of their major challenges. As showed in Figure 2, the techniques commonly used for implementing the different stages of a typical CBR system include: knowledge-based systems, artificial neural networks (ANN), genetic algorithms (GA), rule-based systems (RBS), qualitative reasoning (QR), fuzzy systems (FS) and constraint satisfaction problems (CSP).

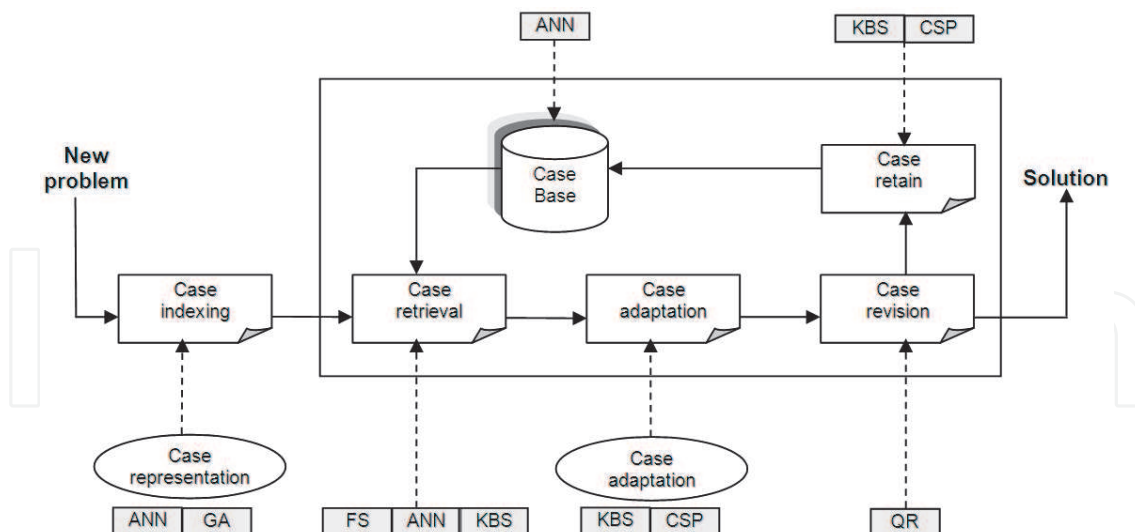


Fig. 2. Methods and techniques commonly used for implementing each stage of a typical CBR system

As the core of CBR systems is its memory, cases should therefore accurately represent both problems and their solutions. In this context, cases may be deleted if they are found to produce inaccurate solutions, they may be merged together to create more generalised solutions, and they may be modified, over time, through the experience gained in producing improved solutions. If an attempt to solve a problem fails and it is possible to identify the reason for the failure, then this information should also be stored in order to avoid the same mistake in the future. This corresponds to a common learning strategy employed in human problem-solving. Rather than creating general relationships between problem descriptors and conclusions, as is the case with rule-based reasoning, or relying on general knowledge of the problem domain, CBR systems are able to utilise the specific knowledge of previously experienced, concrete problem situations. A CBR system provides an incremental learning process because each time a problem is solved, a new experience is retained, thus making it available for future reuse.

As design methodology adequate for constructing DSS, case-based reasoning can be used by itself or as part of another intelligent or conventional computing system. From the last years, there has been an increasing interest in the possibility of integrating different AI techniques with the goal of constructing more powerful and accurate hybrid systems. In this context, the work of Soucek (1991) established the IRIS (*Integration of Reasoning, Informing and Serving*) classification with the goal of facilitating the efficient design of intelligent systems. In the same line, the work of Medsker & Bailey (1992) proposed five integration models based on symbolic and connectionistic systems. Finally, the work of Bezdek (1994) suggested the CIC (*Computational Intelligence Classification*) schema, an interesting classification guidelines for cataloguing hybrid AI systems.

Therefore, given their hybrid nature, CBR systems can be easily combined with other alternatives in order to construct robust decision support systems. These approaches include their successful hybridization with expert systems (Vo & Macchion, 1993; Rissland et al. 1993; Medsker, 1995), fuzzy logic (Xu, 1994; Gui, 1993; Dubois et al. 1997), genetic algorithms (Louis et al. 1993; Oppacher & Deugo, 1991), qualitative reasoning (Navinchandra et al. 1991), constraint satisfaction systems (Maher & Zhang, 1993; Hinrichs, 1992), artificial neural networks (Thrift, 1989; Lim et al. 1991; Liu & Yan, 1997; Corchado et al. 2001) and bayesian

networks (Shinmori, 1998; Aamodt y Langseth, 1998; Dingsøy, 1998; Frieze, 1999; Langseth et al., 1999) among others.

Furthermore, case-based reasoning can be a particularly appropriate problem-solving strategy when the knowledge required to formulate a rule-based model of the domain is difficult to obtain, or when the number or complexity of rules relating to the problem domain is too large for conventional knowledge acquisition methods. In this sense, according to the work of Aamodt and Plaza (1994) there are five different types of CBR systems, and although they share similar features, each of them is more appropriate for a particular type of problem: (i) exemplar based reasoning (EBR), (ii) instance based reasoning (IBR), (iii) memory-based reasoning (MBR), (iv) analogy-based reasoning (ABR) and (v) typical case-based reasoning (CBR).

EBR systems are especially suitable for classification tasks, in which the category of the most similar past case becomes the solution to the new problem. The set of classes directly constitutes the collection of possible solutions applied without modification. IBR systems are a specialization of exemplar-based reasoning for solving problems in which the instances (cases) are usually very simple (e.g. feature vectors). These systems can be completely automated with no user intervention in their whole life cycle. MBR systems supplement previous approaches with the capacity of parallel processing computation. ABR systems are particularly applicable for solving new problems based on past cases from a different domain. In order to properly work, it should be possible to transfer the solution of a source analogue situation to the present target problem. In typical CBR systems cases are assumed to present a certain complexity level (in terms of their internal structure and the information contained), therefore some modification is needed in order to adapt retrieved solutions when applied to a different problem solving context.

### 3. Practical applications

The decision support systems covered in this chapter come from four different research areas: industrial planning, biomedical domain, oceanographic forecasting and anti-spam filtering. All the implemented applications are fully designed following the CBR paradigm in order to empower their adaptability and accuracy for solving new problems in their respective fields.

For each domain, we first introduce the target problem to be solved together with the main aspects surrounding each particular situation. A clear description of the representation used for defining the case base is presented and the internal architecture governing each system is explained in detail.

#### 3.1 Industrial planning

Production scheduling is one of the most important functions in a production company. As a consequence, in recent decades various methods have been proposed for the modelling and solution of particular scheduling problems (Akyol & Bayhan, 2007). In the particular case of cooperative poultry farms, the accurate coordination of centralized feed supply (production and distribution) between scattered farms is of utmost importance for both the main feed manufacturer and participating farmers.

In such a situation, some key aspects involving the main participants need to be taken into consideration, for example (i) the feed production plant has a limited production and storage capacity, (ii) the plant manufactures several types of feed that can create resource



conflicts in production (i.e. they can not be treated as one single product) and distribution (i.e. they can not share the same storage areas in vehicles), (iii) deliveries to farmers can be made in advance, but not late, (iv) each farm has a limited storage area leading to inventory holding costs, (v) some vehicles can not access certain farms due to their dimensions, etc. The problem is difficult since it combines two sub-problems known to be NP-hard: a multi-product, multi-period production problem and a split delivery periodic vehicle routing problem (Boudia, 2008).

In this context, a worthwhile objective for the main feed production plant of any cooperative is to determine, on a daily basis, (i) the amount produced for each type of feed, (ii) the quantities of feed distributed to each farmer and (iii) the associated delivery trips to minimize total cost over the horizon (setup, holding and distribution costs). In this line we present the SP4 (*System for Prediction & Planning of Provision Production*) application, a decision support system that combines a statistical method (used to calculate the previous consumption data, mortality indices and feed delivery types), a machine learning method (M5 Prime and IBk models - used to calculate the total amount of feed consumed by type) and an ad-hoc algorithm which makes flexible orders for compound feed production forecasting (Reboiro-Jato et al. 2011).

Attribute	Type	Description
Farm Id	Alphanumeric	Farm identifier.
Shed Id	Numeric	Shed identifier. Each farm may have several sheds. This value differentiates each shed in a given farm.
Entry date	dd-mm-yyyy	Date on which the lot arrives at the farm.
Final date	dd-mm-yyyy	Date on which the lot is slaughtered.
Feed type	Enumeration	Identifies the specific variety of feed.
Breed	Enumeration	Specific breed type of the lot.
Activity	Enumeration	This variable indentifies the main role of the lot. There are five possible values: <i>laying hen raising</i> , <i>breeder raising</i> , <i>laying hen laying</i> , <i>breeder laying</i> and <i>fattening</i> .
Subactivity	Enumeration	Activity specialization. Each activity may have several specific variants. Possible values include: <i>broiler</i> , <i>label</i> , <i>light</i> , <i>country</i> , etc.
Type	Enumeration	Class of fatten chicken. Animals used on fatten activity are classified into several categories like 'A', 'C', 'CA', 'H', 'M', etc. This attribute is only used on lots belonging to <i>fatten</i> activity.
Number of males	Numeric	Number of male animals the lot.
Number of females	Numeric	Number of female animals the lot.
Number of unsexed	Numeric	Number of animals for which the sex is unknown.
Season	Enumeration	Period of the year in which the lot spends the most days on the farm.
Kilograms	Numeric	Amount of feed consumed by the lot from its entry to its slaughter.

Table 1. Variables defining the SP4 internal case base

Efficient farming of animal husbandry on such an industrial scale relies largely on several variables and limitations that are normally correlated. In such a situation, characterized by multiple scattered farms that depend on a centralized supply centre, there are two main subjects to take into account: (i) different breeds of poultry growing in the same farm (many animals) with a distinct raise program and (ii) different types of feed needed for each lot in each farm for a given period of time. In order to build an appropriate knowledge base for the accurate operation of the SP4 system, several economic, production and management company databases were studied and pre-processed. The main objective was gathering and filtering real data in order to obtain a valid history of feed orders sent to farmers during recent years. Table 1 presents the final structure of the cases comprising the memory of the SP4 system. The general criterion applied for the development of the SP4 internal case base was to select those orders sent to the farmer in the period comprising four days before each lot arrived at the farm, until its exit. During this process, several rules were codified and applied for detecting and solving existing inconsistencies.

Given the large number of variables and rules that play an important role in the whole process of correctly supplying feed for all the farms comprising a given cooperative, the proposed decision support system has been organized into three different but complementary subsystems: (i) data gathering and feed type identification, (ii) consumption forecasting and (iii) production planning. The developed system is going to execute subsystems one and two in a sequential fashion for each new lot of animals that arrives to a given farm. Finally, subsystem three is responsible for combining all this information on a daily basis and determining the amount of feed that must be produced by day and type to the main feed production plant. Figure 3 depicts the general architecture of the SP4 system.

As it can be seen from the operational diagram showed in Figure 3, each subsystem works with unrelated information coming from both existing data and knowledge bases, but also considering the output of the previous stage. In the proposed architecture, each phase solves a different problem following a bottom-up approach.

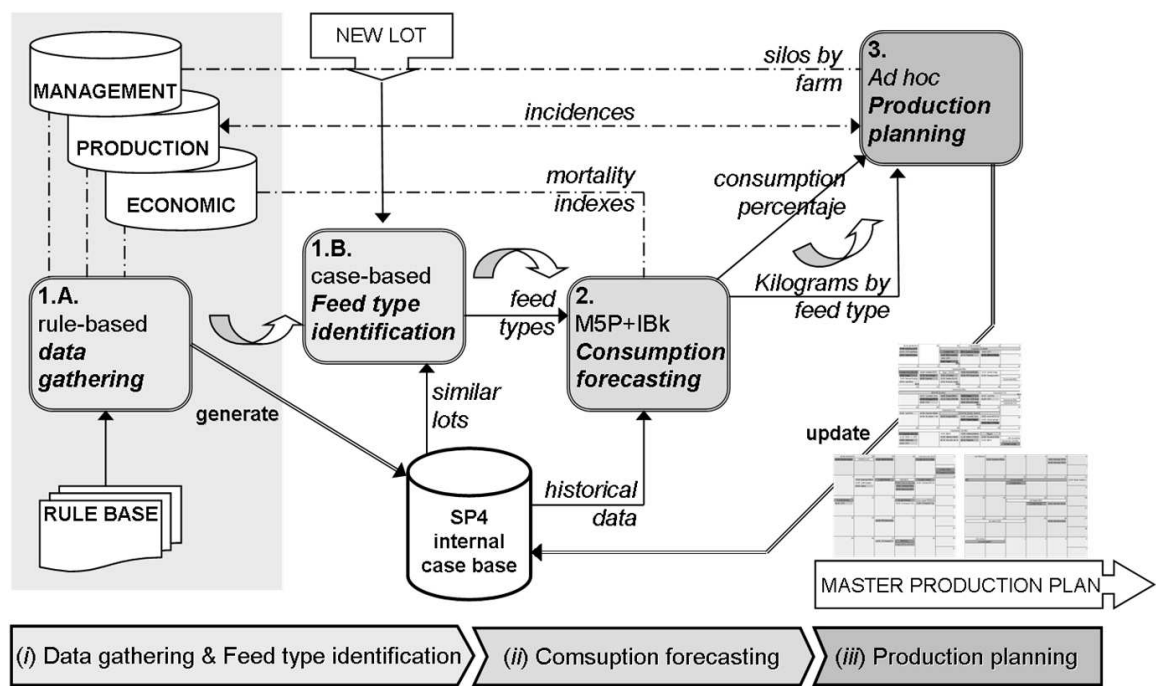


Fig. 3. General architecture and life cycle of the SP4 system for compound feed forecasting and planning

*Subsystem one:* each lot of animals consumes several types of feed during each growth stage. The objective of this phase is to identify the variants of feed that a given lot is going to consume before slaughter, taking into consideration similar past cases belonging to different farms and periods. The current output (qualitative information) serves as the basis for the next phase to compute the total amount of feed needed by both a given lot and a given feed type (quantitative information).

*Subsystem two:* given a new lot and the different types of feed that it will consume before slaughter, it estimates the total consumption (kilograms) by feed type together with the consumption percentage per animal. Several variables influence the output of this phase: number of animals by sex (male and female), week of year, mortality indexes, etc.

*Subsystem three:* this integrates previous calculations by using an ad hoc planning algorithm for simulating feed consumption and production along with the time. It allows the user to easily visualize production activity though an intuitive and configurable GUI (*Graphic User Interface*). Incidences in the initial estimated MPS can be reported and accounted for in reconsidering daily production.

The data used for constructed the system case base was provided by a leading Spanish company specialized in animal feed production and delivery. Raw data (from the years 2007 and 2008) was built from client orders, company production logs, information about the number of animals at different farms and truck trips to the clients. A total of 5112 records were stored in the SP4 internal case base after executing the initial operation for detecting and solving existing inconsistencies (procedure 1.A. in Figure 3). In order to increase the confidence of experimental findings, a cross-validation process was carried out in the all of the experiments.

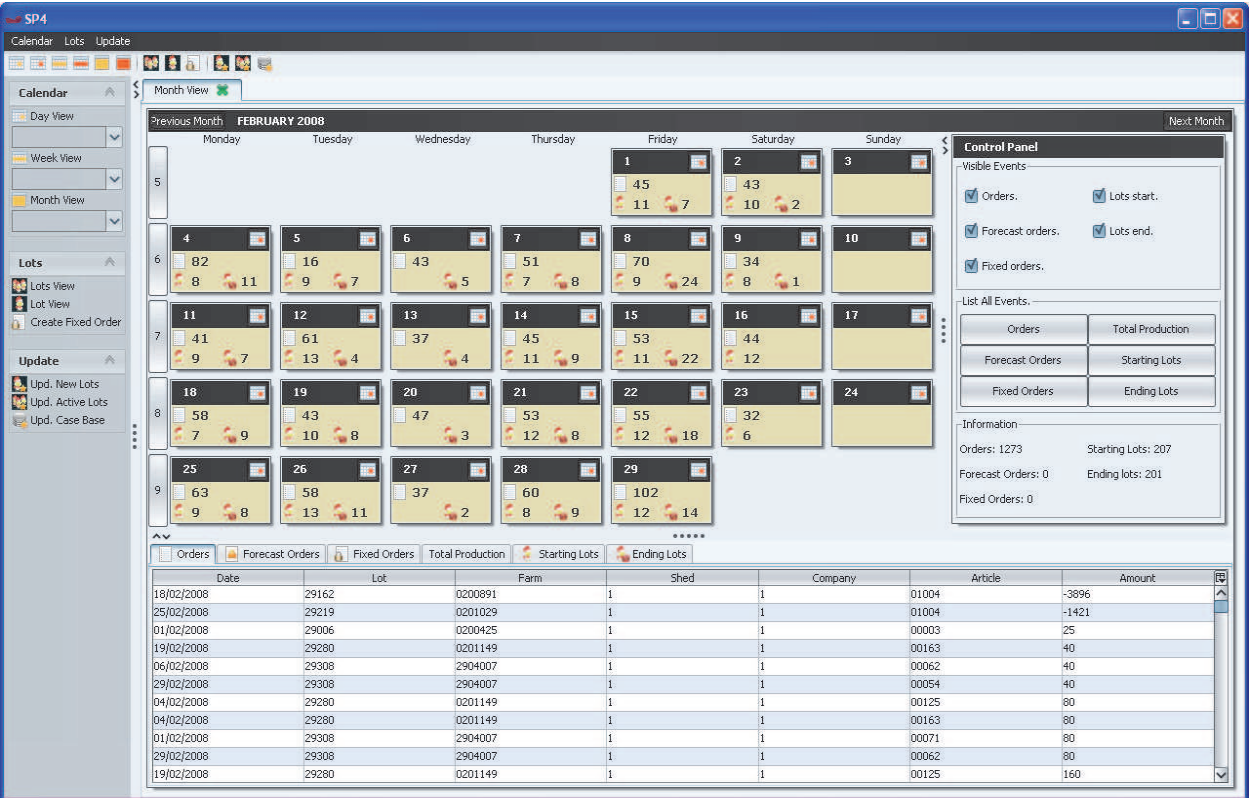


Fig. 4. Screenshot of the final deployed SP4 system showing the Master Production Plan



The final developed system used in the experiments carried out (see Figure 4) was set up to forecast the total amount (kilograms) of each feed type consumed by each lot from its entry date to its slaughter. Moreover, further experiments have been carried out to compare the performance of the SP4 system with several other forecasting approaches. These include standard statistical forecasting algorithms, decision trees and the application of neural networks methods. The results obtained from experimentation revealed that the proposed system performed optimally, being able to track the dynamic non-linear trend and seasonality, as well as the numerous interactions between correlated variables.

### 3.2 Biomedical domain

In recent years DNA microarray technology has become a fundamental tool in genomic research, making it possible to investigate global gene expression in all aspects of human disease (Russo et al. 2003). Microarray technology is based on a database of over 40,000 fragments of genes called expressed sequence tags (ESTs), which are used to measure target abundance using the scanned intensities of fluorescence from tagged molecules hybridized to ESTs. Following the advent of high-throughput microarray technology it is now possible to simultaneously monitor the expression levels of thousands of genes during important biological processes and across collections of related samples. Since the number of examined genes in an experiment runs to the thousands, different data mining techniques have been intensively used to analyze and discover knowledge from gene expression data (Piatetsky-Shapiro & Tamayo 2003). However, having so many fields relative to so few samples creates a high likelihood of finding false positives. This problem is increased if we consider the potential errors that can be present in microarray data.

Bioinformatics and medical informatics are two research fields that serve the needs of different but related communities. Both domains share the common goal of providing new algorithms, methods and technological solutions to biomedical research, and contributing to the treatment and cure of diseases. Although different microarray techniques have been successfully used to investigate useful information for cancer diagnosis at the gene expression level, the true integration of existing methods into day-to-day clinical practice is still a long way off (Sittig et al. 2008). Within this context, case-based reasoning emerges as a suitable paradigm specially intended for the development of biomedical informatics applications and decision support systems, given the support and collaboration involved in such a translational development (Jurisica & Glasgow, 2004).

In addressing the issue of bridging the existing gap between biomedical researchers and clinicians who work in the domain of cancer diagnosis, prognosis and treatment using microarray data, we have developed and made accessible a common interactive framework: the geneCBR decision support system (Glez-Peña et al. 2009a). Our geneCBR system implements a freely available software tool that allows the use of combined techniques that can be applied to gene selection, clustering, knowledge extraction and prediction for aiding diagnosis in cancer research. For biomedical researches, geneCBR *expert mode* offers a core workbench for designing and testing new techniques and experiments. For pathologists or oncologists, geneCBR *diagnostic mode* implements an effective and reliable system that can diagnose cancer subtypes based on the analysis of microarray data using CBR architecture. For programmers, geneCBR *programming mode* includes an advanced edition module for run-time modification of previous coded techniques.

In order to initially construct the knowledge base starting from the available patient's data showed in Table 2, geneCBR stores the gene expression levels of each microarray sample in its case base (lower part of Figure 5).

Attribute	Type	Description
NAME	Alphanumeric	Unique identifier. Denomination of the microarray sample.
#Age	Numeric	Age of the patient.
#Sex	Enumeration	Possible values are: <i>male, female</i> .
#FAB/WHO	Alphanumeric	The FAB classification is a morphological characterization of leukemia. The WHO classification incorporates the results of chromosome and genetic research developed during the past 25 years after the FAB classification.
#FISH studies	Alphanumeric	FISH studies are used to delineate complex chromosome rearrangements, diagnose microdeletion syndromes, or demonstrate the presence of molecular rearrangements characteristic of certain hematologic malignancies.
Gene name 1	Alphanumeric	Human gene name or identifier.
Gene value 1	Numeric	Microarray gene expression value for the associated gene identifier.
....	...	The number of gene name-value pairs depends on the microarray type (Affymetrix HG-U133A/B/Plus, etc.).
Gene name <i>n</i>	Alphanumeric	Human gene name or identifier.
Gene value <i>n</i>	Numeric	Microarray gene expression value for the associated gene identifier.
Class	Alphanumeric	Type of disease.

Table 2. Internal representation of a microarray sample in the geneCBR system (the symbol ‘#’ represents an optional feature)

During the retrieval stage, the original case vectors are transformed into fuzzy microarray descriptors (FMDs). Each FMD is a comprehensible descriptor of the sample in terms of a linguistic label for each gene expression level (central part of Figure 5). This transformation is carried out by means of an accurate fuzzy discretization process (Díaz et al. 2006). Based on the FMD representation created from the case base, a set of fuzzy patterns (FP) is constructed that represents the main characteristics of the a priori known classes (top-left square in Figure 5). Each class in the system is then represented by a FP that holds the fuzzy codification of gene expression levels for those genes that were flagged as relevant for this class. Several FPs are generated from the data in a supervised way, each one representing a group of FMDs for pathological (or generic) specy.

The retrieval stage in the geneCBR system uses the FPs in order to select the most representative genes given a new patient. This phase can be thought of as a gene selection step, in which the aim is to retrieve the list of genes that might be most informative given a new sample to classify. Since it is highly unlikely that all the genes have significant information related to cancer classification and the dimensionality would be too great if all the genes were used, it is necessary to explore an efficient way to obtain the most suitable group of genes. In order to make this selection, our geneCBR system selects those fuzzy patterns from its case base which are the nearest to any new case obtained. Then, for each one of the selected FPs, the geneCBR system computes its associated DFP (a pattern which only includes the genes that are necessary in order to discriminate the novel instance from other different classes). Finally, the selected genes for the new case are obtained by joining together the genes belonging to the DFPs considered.

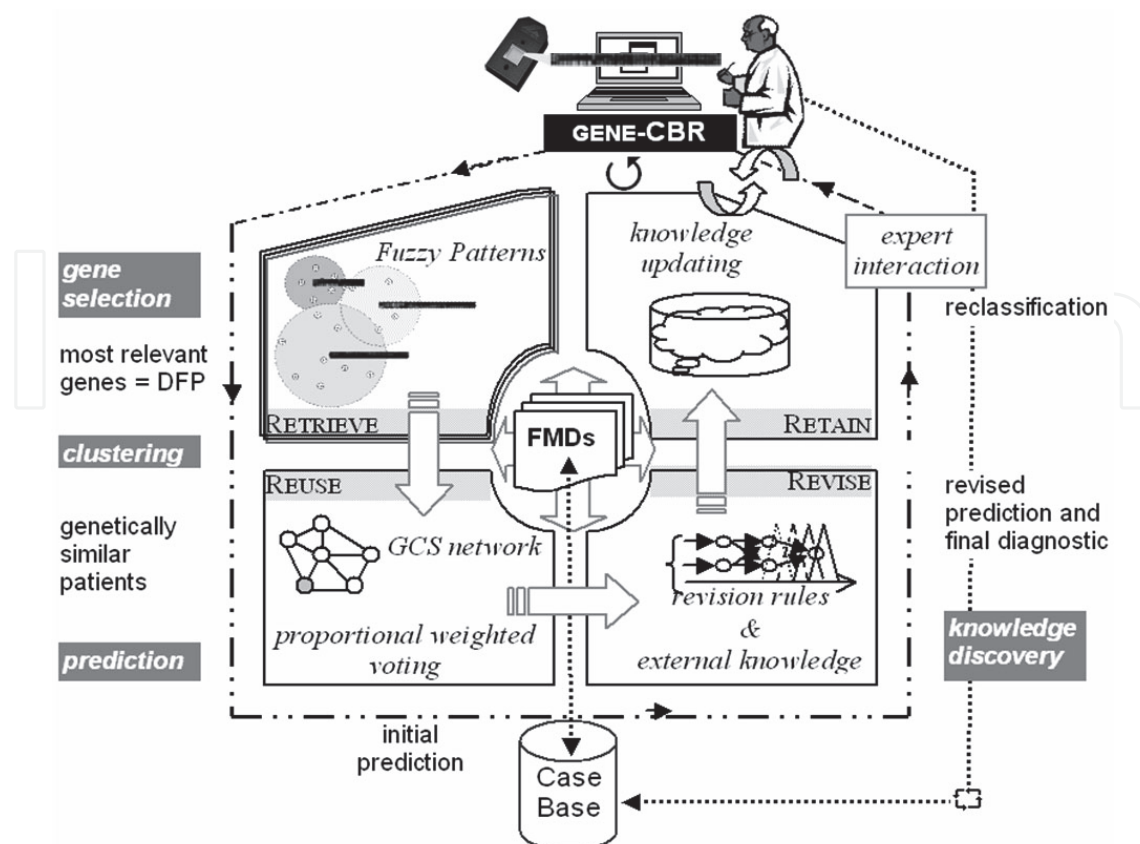


Fig. 5. Life cycle of the geneCBR system working in *diagnostic mode*

The adaptation of previous cases in order to solve a new FMD is accomplished in the reuse stage (left bottom square in Figure 5). A growing cell structures (GCS) network (Fritzke, 1993) is trained with the whole case base, only taking the existing cases represented by the genes selected in the previous stage as input. Then, the new FMD is presented to the GCS network and the patients most similar from a genetic point of view are retrieved. Based on this grouping, a proportional weighted voting mechanism is applied that ponders the level of similarity with the new FMD. An initial class is assigned by the geneCBR system from among the existing pathologies (Glez-Peña et al. 2009b).

In the revision stage (right bottom square in Figure 5) the expert is provided with useful data about the decision made by the system. This information contains the selected DFP genes, the grouping made by the GCS network, the weighting assigned to each class and a set of See5 (Quinlan, 2000) classification rules generated from the most similar patients. The expert contrasts the initial prediction given by the system with other external information such as patient karyotype or clinical history in order to ascertain a revised prediction and a final diagnostic.

Every time a new FMD is solved, the internal structure of the geneCBR system is updated (top-right square in Figure 5). The new FMD is associated to its corresponding class and added to the case base. The affected FP is updated and the system marks the most similar patients selected for future classifications. In this stage the geneCBR system changes to edit-mode and the expert is permitted to update patient's classification taking into account the new knowledge generated.

The geneCBR represents a new translational decision support system that can effectively support the integrative work of programmers, biomedical researches and clinicians working

together in a common framework. Figure 6 shows a screenshot of the geneCBR system working in expert mode (specially intended for biomedical researches). The code of the project is freely available under the GPL license and can be obtained at <http://www.geneabr.org/>.

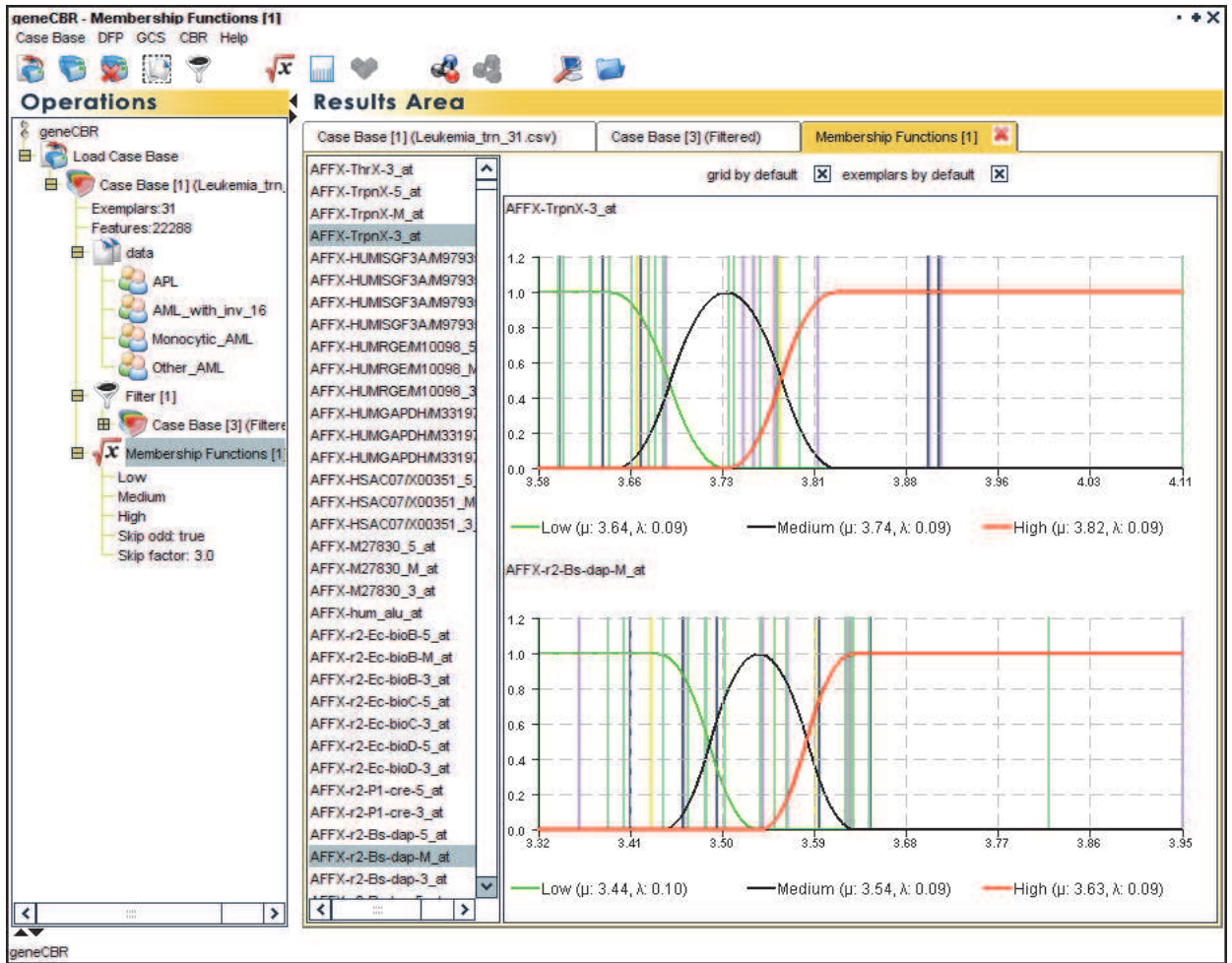


Fig. 6. Screenshot of the geneCBR system working in expert mode

3.3 Oceanographic forecasting

The oceans of the world form a highly dynamic system for which it is difficult to create mathematical models (Tomczak & Fodfrey, 1994). *Red tides* are the name for the discolourations caused by dense concentrations of microscopic sea plants, known as phytoplankton. The discolouration varies with the species of phytoplankton, its pigments, size and concentration, the time of day, the angle of the sun and other factors. Red tides usually occur along the North West coast of the Iberian Peninsula in late summer and autumn. The prevailing southerly winds cause cold, nutrient-rich water to rise up from the deeper regions of the ocean to the surface, a process known as upwelling. Swept along with this upwelled water are dinoflagellate cysts, the resting stages of the organism, which lie dormant in the sediments on the sea floor. The high nutrient concentrations in the upwelled water, together with ideal conditions of temperature, salinity and light, trigger the germination of the cysts, so that the dinoflagellates begin to grow and divide. The rapid increase in dinoflagellate numbers, sometimes to millions of cells per liter of water, is described as a bloom of phytoplankton (concentration levels above the 100,000 cells/liter).



Concentration of the bloom by wind and currents, as well as the dinoflagellates’ ability to swim to the surface, combines to form a red tide.

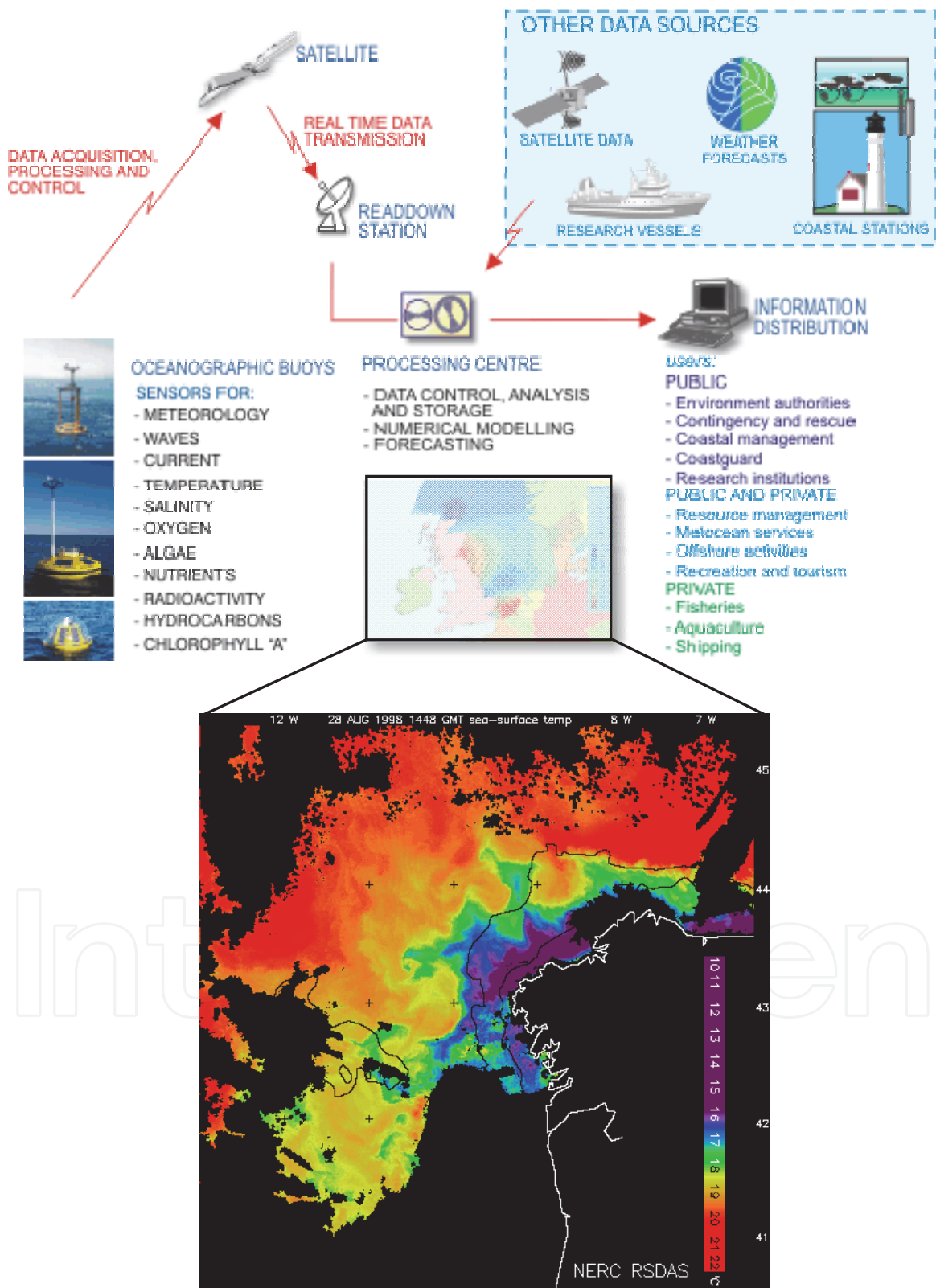


Fig. 7. Data sources used by the FSfRT system



In this context, two situations of special interest are those corresponding to the *false alarms* and the *blooms not detected*. The former refers to predictions of bloom (concentration of pseudo-nitzschia  $\geq 100,000$  cell/liter) which do not actually materialize (real concentration  $\leq 100,000$  cell/liter). The latter, more problematic, occurs when a bloom exists but the model fails to detect it. Another unwelcome situation takes place when the number of predictions exceeds an absolute error of 100,000 cell/liter (labelled as incorrect predictions). In such a situation, in which the rules governing a target process or system are unknown, the prediction of the parameter values that determine the characteristic behaviour of the system can be a problematic task. However, it has been found that a hybrid case-based reasoning system can provide a more effective means of performing such predictions than other connectionist or symbolic techniques (Fdez-Riverola & Corchado, 2003).

Related with this domain we describe the FSfRT (*Forecasting System for Red Tides*) system, a hybrid model able to accurately forecast the concentrations of pseudo-nitzschia spp, the diatom that produces the most harmful red tides causing amnesic shellfish poisoning (or ASP). Our FSfRT system employs a case-based reasoning model to wrap a growing cell structures network, a radial basis function network (Fritzke, 1994) and a set of Sugeno fuzzy models (Jang et al. 1997) to provide an accurate prediction. Each of these techniques is used at a different stage of the reasoning cycle of the decision support system to retrieve historical data, to adapt it to the present problem and to automatically review the proposed solution.

The forecasting system uses information from two main sources: (i) data coming from several buoys and monitoring net used to create a succession of problem descriptors able to characterize the current forecasting situation and (ii) data derived from satellite images stored on a database. The satellite image data values are used to generate cloud and superficial temperature indices which are then stored with the problem descriptor and subsequently updated during the CBR operation. Figure 7 shows a schematic view of the whole data managed by the FSfRT system.

In order to forecast the concentration of pseudo-nitzschia spp at a given point a week in advance, a problem descriptor is generated on a weekly basis. A problem descriptor consists of a sequence of sampled data values (filtered and pre-processed) recorded from the water mass to which the forecast will be applied. The problem descriptor also contains various other numerical values, including the current geographical location of the sensor buoys and the collection time and date. Every week, the concentration of pseudo-nitzschia spp is added to a problem descriptor forming a new input vector. The problem descriptor is composed of a vector with the variables that characterise the problem recorded over two weeks. The prediction or output of the system is the concentration of pseudo-nitzschia spp one week later, as indicated in Table 3.

The cycle of forecasting operations (which is repeated every week) proceeds as depicted in Figure 8. First a new problem instance is created from the pre-processed data cited above. When a new problem is presented to the system, the GCS neuronal network is used to obtain  $k$  more similar cases to the given problem (identifying the class to which the problem belongs). In the reuse phase, the values of the weights and centers of the neural network used in the previous forecast are retrieved from the knowledge base. These network parameters together with the  $k$  retrieved cases are then used to retrain the RBF network and to obtain an initial forecast of the concentration of pseudo-nitzschia spp (see Figure 8). During this process the values of the parameters that characterise the network are updated.

Attribute	Type	Description
Location	Alphanumeric	Geographical location of the sensor buoy.
Date	dd-mm-yyyy	Date on which the measure was made.
Time	hh-mm-ss	Time on which the measure was made.
Temperature	Numeric	Water temperature (cent. degrees) at different depths.
Oxygen	Numeric	Oxygen concentration (milliliters/liter) at different depths.
PH	Numeric	acid/based scale.
Transmittance	Numeric	Fraction (percentage) of sun light that passes through the sea-water.
Fluorescence	Numeric	Sea-water fluorescence (percentage) at different depths.
Cloud index	Numeric	Cloud measurement derivate from a geostationary satellite.
Recount of diatoms	Numeric	Algae concentration (in cell/liter) at different depths.
Pseudo-nitzschia spp	Numeric	Diatom concentration (in cell/liter) at different depths causing harmful algae blooms.
Pseudo-nitzschia spp (future)	Numeric	Diatom concentration (in cell/liter) to be predicted.

Table 3. Oceanographic parameters and physical characteristics of the water mass comprising a case in the FSfRT system

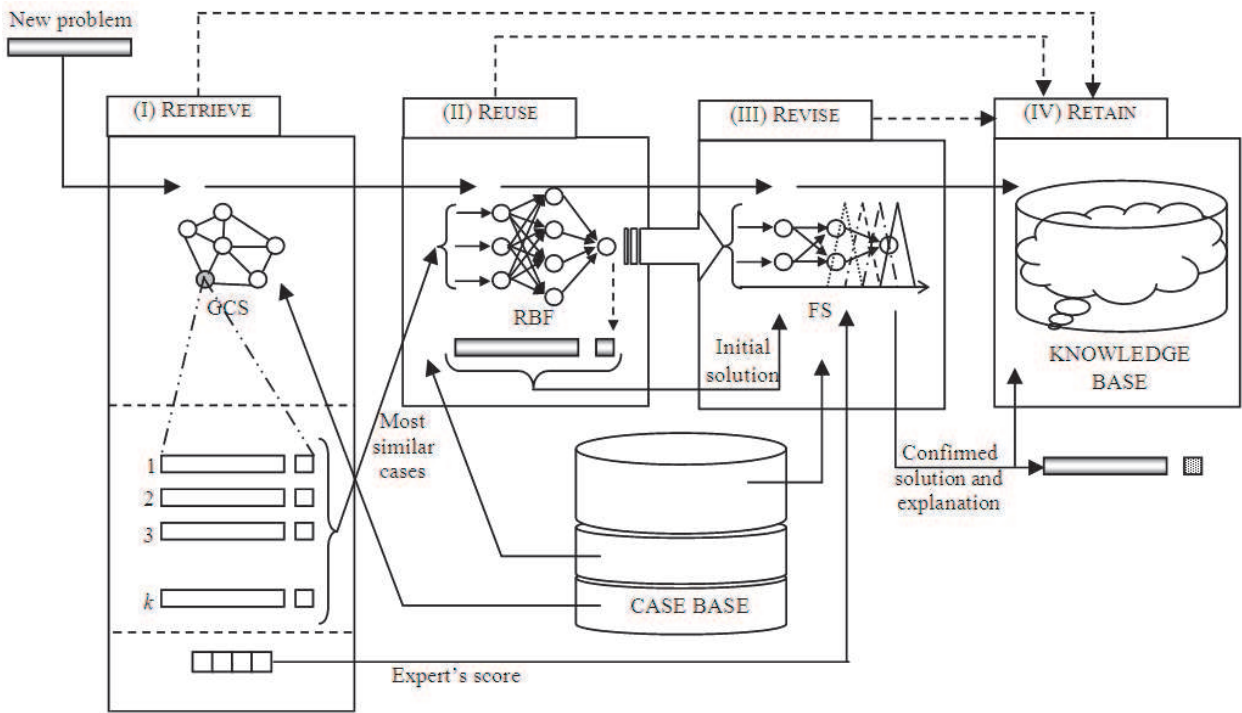


Fig. 8. Life cycle of the FSfRT system for oceanographic prediction

In the revision phase, the initial solution proposed by the RBF neural network is modified according to the responses of the four fuzzy revision subsystems. Each revision subsystem has been created from the RBF network using neurofuzzy techniques (Jin & Sendhoff, 2003). For each class of the GCS neural network a vector of four values is maintained (see Figure

8). This *expert's score* vector is initialised with a value of (0.25, 0.25, 0.25, 0.25) and represents the accuracy of each revision subsystem with respect to a class. During revision, the appropriate *expert's score* vector is used to ponder the outputs of each fuzzy revision system. Each vector value is associated with one of the four revision subsystems. For each forecasting cycle, the value of the importance vector associated to the most accurate revision subsystem is increased and the other three values are proportionally decreased. This is done in order to give more relevance to the most accurate revision subsystem.

The revised forecast is then retained temporarily in the forecast database. When the real value of the concentration of pseudo nitzschia spp is measured, the forecast value for the variable can then be evaluated, though comparison of the actual and forecast value and the error obtained. A new case, corresponding to this forecasting operation, is then stored in the case base. The forecasting error value is also used to update the importance vector associated with the revision subsystems of the retrieved class.

The FSfRT system was successfully tested using real data collected from years [1992, 2000] coming from geographical area A0 (42°28.90' N, 8°57.80' W 61 m). Figure 9 shows a screenshot of the FSfRT interface implemented for oceanographic forecasting.

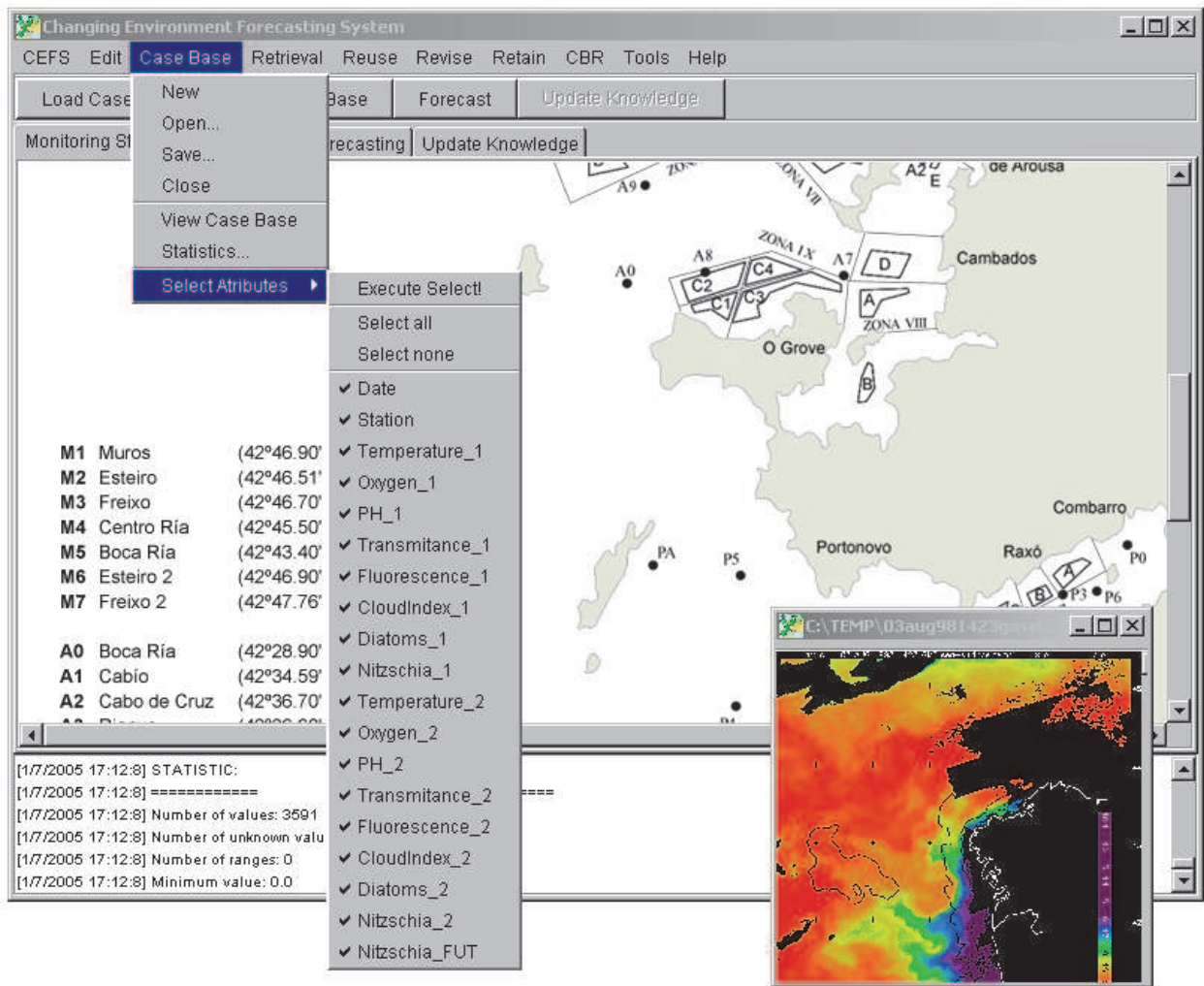


Fig. 9. Screenshot of the FSfRT system forecasting red tides in the coastal waters of the North West of the Iberian Peninsula

### 3.4 Anti-spam filtering

The E-mail service is a computer-based technology built as the result of transforming the old postal delivery in order to use it over networks and Internet. Nowadays, e-mail addresses are present on every business card close to other relevant contact info such as the postal address or the phone number. However, for more than one decade the use of e-mail has been bedeviled by the curse of spamming, so spam is beginning to undermine the integrity of e-mail and even to discourage its use.

In this context, spam is a term used to designate all forms of unsolicited commercial e-mail and can be formally defined as an electronic message satisfying the following two conditions: (i) the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients and (ii) the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent (SpamHaus, 1998).

Due to some attractive characteristics of e-mail (low cost & fast delivery) it actually becomes the main distribution channel of spam contents. Every day e-mail users receive lots of messages containing offers to buy illegal drugs, replicas of Swiss watches, fake jobs, forged university diplomas, etc. This situation has led to a progressive increasing of the spam global ratio in email traffic. During September 2010, the percentage of spam deliveries accounted for about 92 percent of all Internet e-mail traffic (MessageLabs, 2010).

In order to successfully fight against spam (i.e. ideally eliminate it), both theoretical and applied research on spam filtering becomes fundamental. In this context, much valuable research work has been previously carried out (Guzella & Caminhas, 2009) and some relevant conferences have grown up in the field (CEAS, 2010). Moreover, several commercial products have been released and distributed from the software industry to a huge amount of final users with the goal of minimizing spam drawbacks.

With the goal of providing an effective solution we present the SpamHunting system (Fdez-Riverola et al. 2007), an instance-based reasoning e-mail filtering model that outperforms classical machine learning techniques and other successful lazy learner's approaches in the domain of anti-spam filtering. The architecture of the decision support filter is based on a tuneable enhanced instance retrieval network able to accurately generalize e-mail representations. The reuse of similar messages is carried out by a simple unanimous voting mechanism to determine whether the target case is spam or not. Previous to the final response of the system, the revision stage is only performed when the assigned class is spam whereby the system employs general knowledge in the form of meta-rules.

In order to correctly represent incoming e-mails, a message descriptor (instance) is generated and stored in the e-mail base of the SpamHunting system. This message descriptor contains the sequence of features that better summarize the information contained in the e-mail. For this purpose, we use data from two main sources: (i) information obtained from the header of the e-mail and (ii) those terms that are more representative of the subject, body and attachments of the message. Table 4 summarizes the structure of each instance stored in the SpamHunting e-mail base.

Figure 10 illustrates the life cycle of the IBR SpamHunting system as well as its integration within a typical user environment. In the upper part of Figure 10, the mail user agent (MUA) and the mail transfer agent (MTA) are in charge of dispatching the requests generated by the user. Between these two applications, SpamHunting captures all the incoming messages (using POP3 protocol) in order to identify, tag and filter spam.



Attribute	Type	Description
ID	Numeric	Unique message identifier.
From	Alphanumeric	Source mailbox.
Return Path	Alphanumeric	Indicates the address that the message will be returned to if one chose to reply.
Date	dd-mm-yyyy	Date in which the message was sent.
Language	Alphanumeric	Particular tongue of the message.
Attached Files	Numeric	Indicates the number of attached files.
Content Type	Enumeration	MIME type.
Relevant Terms	Numeric	Number of selected features to cluster the message.
Total Terms	Numeric	Number of features contained in the message.
Frequency-Term Descriptor	Array of feature-frequency pairs	Storing for each feature a measure of their frequency in the message.
Class	Enumeration	Message category. Possible values are: <i>spam</i> , <i>legitimate</i> , <i>unknown</i> .

Table 4. Structure of an instance representing an incoming e-mail in the SpamHunting system

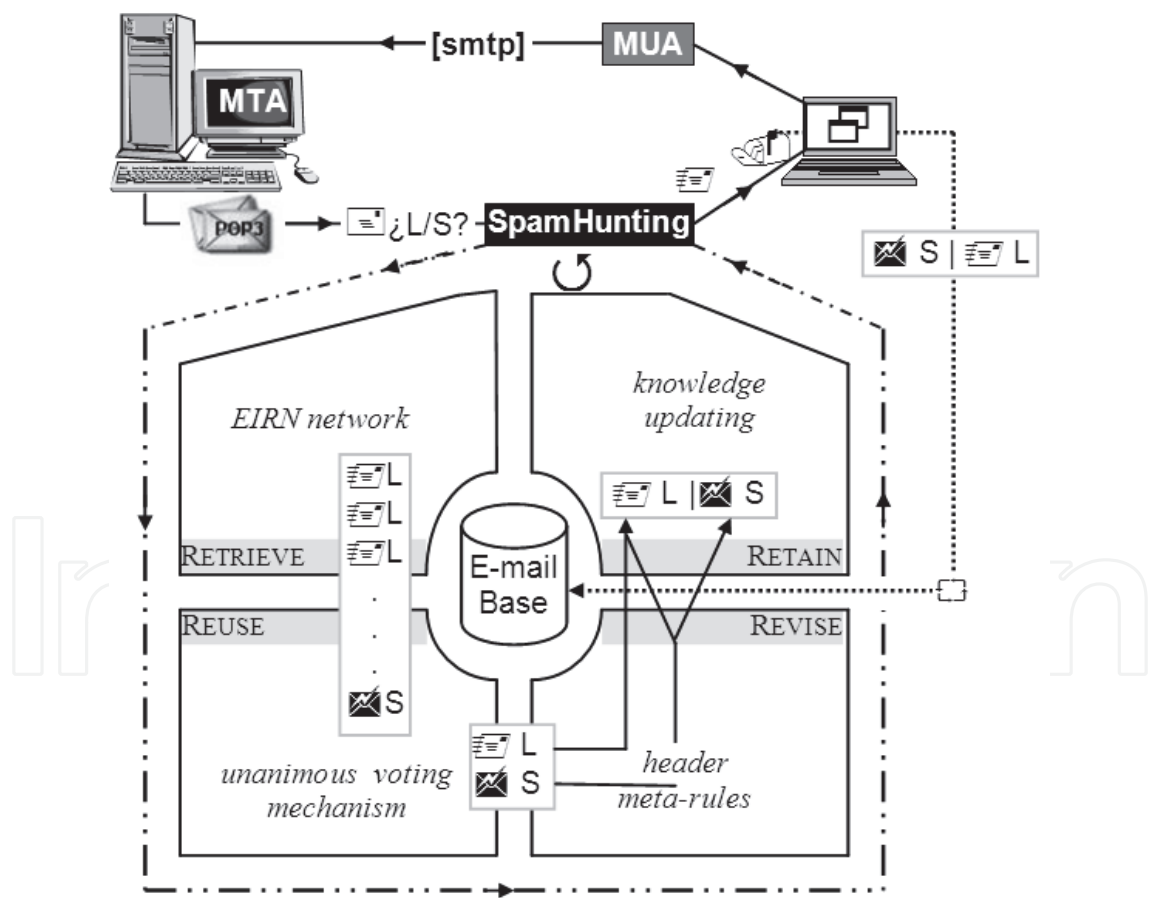


Fig. 10. Life cycle of the SpamHunting system and its integration with existing MTA and MUA

Whenever SpamHunting receives a new e-mail, the system evolves through the four steps depicted in the lower part of Figure 10 as shadowed rectangles. Initially the system



identifies those e-mails that best represent the new incoming message (left upper quadrant in Figure 10), only taking into account the set of messages with the highest number of terms in common. Each one of the previous selected messages contributes with one vote to the final class (left bottom quadrant in Figure 10). The revision stage is only carried out when the system proposes an e-mail as spam. For this purpose, SpamHunting uses previous encoded header meta-rules (right bottom quadrant in Figure 10). Every time the user checks his mailbox and provides feedback to the system about a previous e-mail classification, SpamHunting stores a new instance (or modifies an existing one) in the e-mail base for future use (right upper quadrant in Figure 10).

The retrieval stage is carried out using our enhanced instance retrieval network (EIRN) model. The EIRN network facilitates the indexation of instances and the selection of those that are most similar to the instance-message. The reuse of similar e-mails is carried out by means of the utilization of a unanimous voting mechanism, which generates an initial solution by creating a model with the retrieved instances. In the revision stage the system employs general knowledge in the form of meta-rules that are extracted from the e-mail headers. Finally, the retain (learning) stage is carried out whenever the system classifies an incoming e-mail, updating the knowledge structure of the whole system. The proposed system also takes into account the feedback of the user when it receives an incorrectly classified e-mail (dotted arrow from the user computer to the e-mail base in Figure 10).

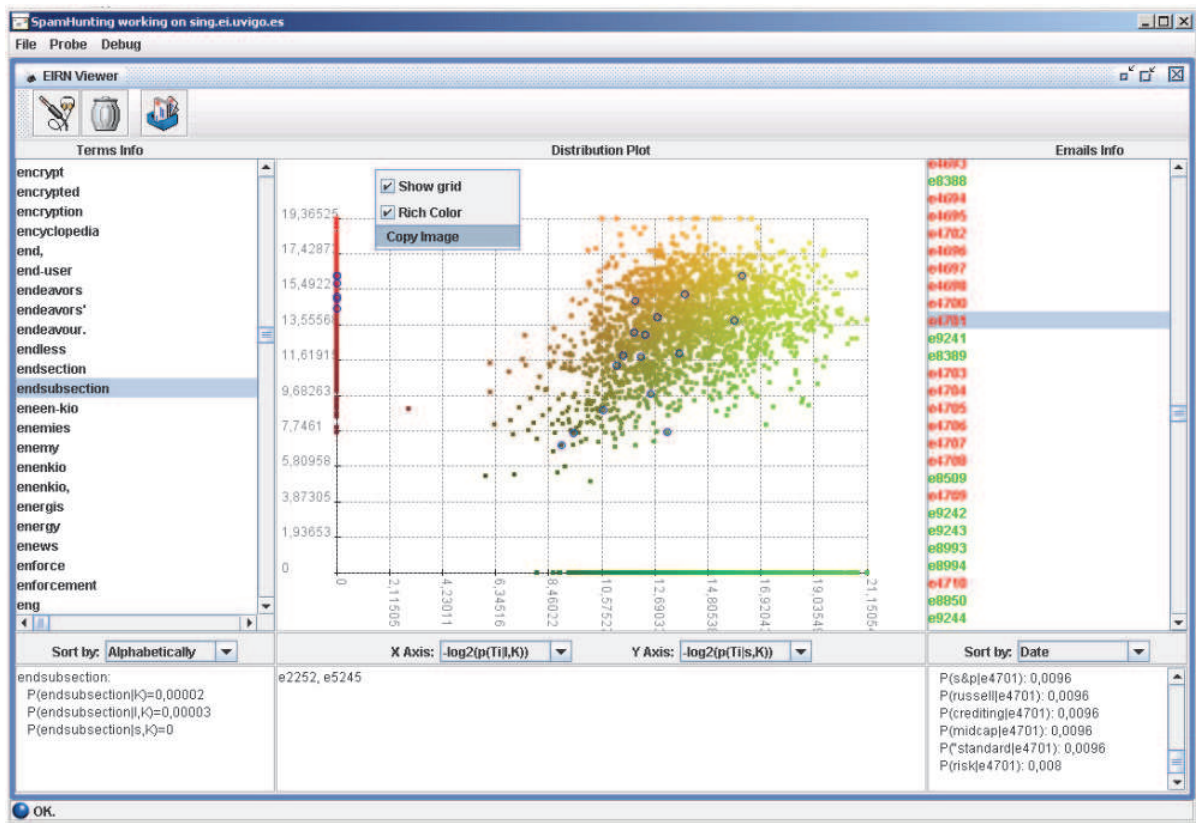


Fig. 11. Screenshot of the EIRN viewer comprising the SpamHunting system

One of the most relevant features of our EIRN structure is the possibility of generating different views of the underlying indexed knowledge. Figure 11 shows a screenshot of the software developed for taking advantage of this capacity. As we can observe from Figure 11,

the application frame is structured into six panels. First, on the top of the left column, we can find all the relevant terms that the EIRN network uses for indexing existing instances. Under this panel, a section is located that summarizes some statistics referring to the selected terms including: (i) probability of finding the term in the case base (ii) frequency of finding the term in spam e-mails and (iii) probability of finding the term in legitimate messages.

At the top of the right column, we can find all the instances stored in the system memory. Under this panel, an instance statistics panel can be found where all the relevant terms belonging to the selected messages as well as their frequencies are showed.

At the top of the central column, we have placed a plot representing the relevant terms indexed in our EIRN model. The plot panel has been developed with built-in clipboard copy support, save images capability and drag and drop compatibility for use with any image application or word processor. Both selected terms belonging to the left panel or part of a message from the right panel are always highlighted in our graphic representation.

The EIRN viewer module represents each term as a two-dimension point on the plot. The coordinates of each point are computed according to the function selected in the combo boxes placed under the plot. The following measurements are available for each coordinate: (i) the probability of finding the term  $t$  in spam e-mails stored in the system memory,  $p(t | s, K)$ , (ii) the logarithmic form of the previous value,  $-\log_2(p(t | s, K))$  (iii) the probability of finding the term  $t$  in legitimate messages,  $p(t | l, K)$ , (iv) the logarithmic form of the previous value,  $-\log_2(p(t | l, K))$ , and (v) the probability of finding the term  $t$  in the system memory,  $p(t | K)$ .

The results obtained confirm the idea that instance-based reasoning systems can offer a number of advantages in the spam filtering domain. Spam is a disjoint concept and IBR classification works well in this domain. In addition IBR systems can learn over time simply by updating their memory with new instances of spam or legitimate e-mail. Moreover, it provides seamless learning capabilities without the need for a separate learning process and facilitates extending the learning process over different levels of learning. The code can be freely obtained at <http://www.spamhunting.org/>.

## 4. Conclusion

In this chapter we have presented the CBR paradigm as an appropriate methodology for implementing successful decision support systems together with their adequate application to several domains. The adoption of CBR methodology for constructing decision support systems has several remarkable benefits, allowing us to obtain a more general knowledge of the system and to gain a deeper insight into the logical structure of the problem and its solution. Main properties of these systems include (i) their ability to focus on the problem's essential features, (ii) the possibility of solving problems in domains that are only partially understood, (iii) the competence for providing solutions when no algorithmic method is available and (iv) the potential to interpret and manage open-ended and ill-defined concepts.

The main advantages of case-based reasoning paradigm over other alternative approaches for the implementation of decision support systems are related with the fulfilment of the following characteristics: (i) reduce the knowledge acquisition effort (cases are independent from each other), (ii) require less maintenance effort (partially automatically by adding/deleting cases), (iii) improve problem solving performance through reuse, (iv) makes use of existing data (e.g. in databases), (v) improve over time and adapt to changes in the environment and (vi) present high user acceptance (domain experts and novices understand cases quite easy).

However, the effective utilization of case-based reasoning paradigm also presents inherent limitations and/or drawbacks related to its knowledge representation and life cycle: (i) past cases could be inexistent or difficult to represent, (ii) specific techniques have to be defined from scratch for modifying previous cases or their solutions in order to adapt them to the new situations and (iii) in some scenarios, it could be difficult to maintain case-base efficiency because unused cases need to be forgotten. In such situations, specific solutions have to be defined in order to overcome particular difficulties.

## 5. Acknowledgements

This work has been partially funded by the project InNoCBR (10TIC305014PR) from Xunta de Galicia. M. Reboiro-Jato was supported by a pre-doctoral fellowship from University of Vigo.

## 6. References

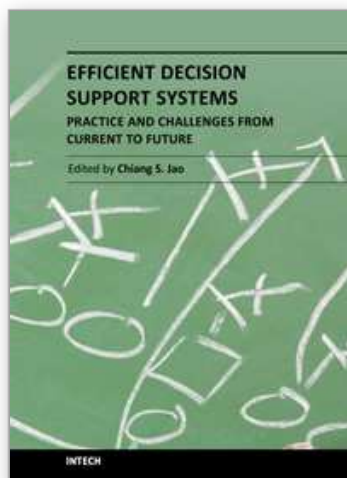
- Aamodt, A. & Langseth, H. (1998). *Integrating Bayesian Networks into Knowledge-Intensive CBR*. Technical Report WS-98-15, AAAI Press, Menlo Park, pp. (1-6)
- Aamodt, A. & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, Vol.7, pp. (39-59)
- Akyol, D.E. & Bayhan, G.M. (2007). A review on evolution of production scheduling with neural networks. *Computers & Industrial Engineering*, Vol.53, No.1, pp. (95-122)
- Bezdek, J.C. (1994). What is Computational Intelligence?. In: *Computational Intelligence: Imitating Life*, Zurada J.M., Marks II, R.J., & Robinson C.J. (Eds.), pp. (1-12), IEEE Press, ISBN 978-0780311046, New York
- Boudia, M. (2008). Coordination of production planning and distribution. *4OR: A Quarterly Journal of Operations Research*, Vol.6, No.1, pp. (93-96)
- CEAS, (2010). *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, Available from <http://www.ceas.cc/>
- Corchado, J.M., Aiken, J. & Rees, N. (2001). *Artificial Intelligence Models for Oceanographic Forecasting*. Plymouth Marine Laboratory, ISBN 0-951-9618-4-5, UK
- Díaz, F., Fdez-Riverola, F. & Corchado, J.M. (2006). geneCBR: a Case-Based Reasoning Tool for Cancer Diagnosis using Microarray Datasets. *Computational Intelligence*, Vol.22, No.3-4, pp. (254-268)
- Dubois, D., Esteva, F., García, P., Godo, L., López de Mántaras, R. & Prade, H. (1997). Fuzzy set-based models in case-based reasoning. *Proceedings of the 2nd International Conference on Case-Based Reasoning*, Providence, Rhode Island, USA, pp. (599-610)
- Dingsøyr, T. (1998). *Retrieval of Cases by using a Bayesian Network*. Technical Report WS-98-15, AAAI Press, Menlo Park, pp. (50-54)
- Fdez-Riverola, F. & Corchado, J.M. (2003). CBR based System for Forecasting Red Tides. *International Journal of Knowledge-Based Systems*, Vol.16, No.5-6, pp. (321-328)
- Fdez-Riverola, F., Iglesias, E.L., Díaz, F., Méndez, J.R. & Corchado, J.M. (2007). SpamHunting: An Instance-Based Reasoning System for Spam Labelling and Filtering. *Decision Support Systems*, Vol.43, No.3, pp. (722-736)
- Friese, T. (1999). Utilization of Bayesian Belief Networks for Explanation-Driven Case-Based Reasoning. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Workshop ML-5: On the automatic construction of case-based reasoners, Stockholm, Sweden, pp. (73-76)

- Fritzke, B. (1993). Growing self-organising Networks – Why?, *Proceedings of the 11th European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp. (61-72)
- Fritzke, B. (1994). Fast learning with incremental RBF Networks. *Neural Processing Letters*, Vol.1, No.1, pp. (2-5)
- Glez-Peña, D., Díaz, F., Hernández, J.M., Corchado, J.M. & Fdez-Riverola, F. (2009a). geneCBR: a translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. *BMC Bioinformatics*. 10:187
- Glez-Peña, D., Díaz, F., Fdez-Riverola, F., Méndez, J.R. & Corchado J.M. (2009b). Fuzzy Patterns and GCS Networks to Clustering Gene Expression Data. In: *Fuzzy Systems in Bioinformatics and Computational Biology. Studies in Fuzziness and Soft Computing*, pp. (103-125), Springer, ISBN: 978-3-540-89967-9, Germany
- Gui, J.K. (1993). Methodology for modelling complete product assemblies. *Act Polytechnica Scandinavica. Mathematics and Computer Science*, Vol.60 pp (1-189)
- Guzella, T.S. & Caminhas, W.M. (2009) A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, Vol.36, No.7, pp. (10206-10222)
- Hinrichs, T.R. (1992). *Problem solving in open worlds: A case study in design*. Lawrence Erlbaum, ISBN 0805812288, Hillsdale, New York
- Jang, J.-S.R., Sun, C.-T. & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, ISBN 978-0132610667, US
- Jin, Y. & Sendhoff, B. (2003). Extracting Interpretable Fuzzy Rules from RBF Neural Networks. *Neural Processing Letters*, Vol.17, No.2, pp. (149-164)
- Jurisica, I. & Glasgow, J. (2004). Applications of case-based reasoning in molecular biology. *Artificial Intelligence Magazine, Special issue on Bioinformatics*, Vol.25, No.1, pp. (85-95)
- Klein, G. & Whitaker, L. (1988). Using analogues to predict and plan. *Proceedings of the first DARPA Workshop on Workshop Case-Based Reasoning*, pp. (224-232)
- Kolodner, J. (1993). *Case-based Reasoning*. Morgan Kaufmann, ISBN 1-55860-237-2, San Mateo, CA
- Langseth, H., Asmodt, A. & Martin, O. (1999). Learning Retrieval Knowledge from Data. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Workshop ML-5: On the automatic construction of case-based reasoners*, Stockholm, Sweden, pp. (77-82)
- Lim, J.H., Tan, A.H., Lui, H.C. & Teh, H.H. (1991). INCIDE: A connectionist case-based diagnostic expert system that learns incrementally. *Proceedings of the International Joint Conference on Neural Networks*, Singapore, pp. (1693-1698)
- Liu, Z.Q. & Yan, F. (1997). Fuzzy neural network in case-based diagnostic system. *IEEE Transactions on Fuzzy Systems*, Vol.5, No.2, pp. (209-222)
- Louis, S., McGraw, G. & Wyckoff, R.O. (1993). Case Based Reasoning assisted explanation of genetic algorithm results. *Journal of Experimental and Theoretical Artificial Intelligence*, Vol.5, No.1, pp. (21-37)
- Maher, M.L. & Zhang, D.M. (1993). CADSYN: A case based design process model. *Artificial Intelligence for Engineering Design, Analysis, and Manufacture*, Vol.7, No.2, pp. (97-110)
- Medsker, L.R. (1995). *Hybrid Intelligent Systems*. Kluwer Academic Publishers, ISBN 9780792395881
- Medsker, L.R., & Bailey, D.L. (1992). Models and guidelines for integrating expert systems and neural networks. In: *Hybrid Architectures for Intelligent Systems*. Kandel, A., &



- Langholz, G. (Eds.), pp. (153-171), CRC Press, ISBN 0-8493-4229-5, Boca Raton, Florida
- MessageLabs. (2010). *MessageLabs Intelligence*, Available from <http://www.messagelabs.co.uk/intelligence.aspx>
- Navinchandra, D., Sycara, K.P. & Narasimhan, S. (1991). A transformational approach to case-based synthesis. *Artificial Intelligence in Engineering, Manufacturing and Design*, Vol.5, No.1, pp. (31-45)
- Oppacher, F. & Deugo, D. (1991). Integrated Case-based Reasoning with genetic algorithms. *Proceedings of the International Symposium on Computational Intelligence*, Milan, Italy, pp. (103-114)
- Pal, S.K., Dillon, T.S. & Yeung, D.S. (2000). Soft Computing in Case Based Reasoning. *International Journal of Intelligent Systems*, Vol.16, No.4
- Piatetsky-Shapiro, G. & Tamayo, P. (2003). Microarray data mining: facing the challenges. *ACM SIGKDD Explorations Newsletter*, Vol.5, No.2, pp. (1-5)
- Quinlan, J.R. (2000). *Data mining tools See5 and C5.0*, Available from: <http://www.rulequest.com/see5-info.html/>
- Reboiro-Jato, M., Glez-Dopazo, J., Glez, D., Laza, R., Gálvez, J.F., Pavón, R., Glez-Peña, D. & Fdez-Riverola, F. (2011). Using Inductive Learning to Assess Compound Feed Production in Cooperative Poultry Farms. *Expert Systems With Applications*, Submitted for Publication
- Riesbeck, C.K. & Schank, R.C. (1989). *Inside Case-based Reasoning*. Lawrence Erlbaum Ass, ISBN 0-89859-767-6, Hillsdale
- Rissland, E.L., Daniels, J.J., Rubinstein, Z.B. & Skalak, D.B. (1993). Case-based diagnostic analysis in a blackboard architecture. *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, DC, pp. (66-72)
- Russo, G., Zegar, C. & Giordano, A. (2003). Advantages and limitations of microarray technology in human cancer. *Oncogene*, Vol.22, No.42, pp. (6497-507)
- Schank, R.C. (1982). *Dynamic Memory*. Cambridge University Press, ISBN 0-521-24856-2, Cambridge, UK
- Shinmori, A. (1998). *A Proposal to Combine Probabilistic Reasoning with Case-Based Retrieval for Software Troubleshooting*. Technical Report WS-98-15, AAAI Press, Menlo Park, pp. (149-154)
- Sittig, D.F., Wright, A., Osherooff, J.A., Middleton, B., Teich, J.M., Ash, J.S., Campbell, E. & Bates, D.W. (2008). Grand challenges in clinical decision support. *Journal of Biomedical Informatics*, Vol.41, No.2, pp. (387-92)
- Soucek, B. & IRIS group. (1991). *Neural and Intelligent Systems Integration - Fifth and Sixth Generation Integrated Reasoning Information Systems*. John Wiley & Sons, ISBN 978-0-471-53676-5, New York
- SpamHaus. (1998). *The SpamHaus Project*, Available from <http://www.spamhaus.org/>
- Thrift, P. (1989). A neural network model for case-based reasoning. *Proceedings of the Second DARPA Workshop on Case-Based Reasoning*, Pensacola Beach, Florida, pp. (334-337)
- Tomczak, M. & Godfrey, J.S. (1994). *Regional Oceanographic: An Introduction*. Pergamon, ISBN 978-0080410203, New York
- Vo, D.P. & Macchion, D. (1993). A use of case-based reasoning technique in building expert systems. *Future Generation Computer systems*, Vol.9, No.4, pp. (311-319)
- Watson, I. (1997). *Applying Case-based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann, ISBN 978-1558604629, San Mateo, CA
- Xu, L.D. (1994). Developing a case-based knowledge system for AIDS prevention. *Expert Systems*, Vol.11, No.4, pp. (237-244)





## **Efficient Decision Support Systems - Practice and Challenges From Current to Future**

Edited by Prof. Chiang Jao

ISBN 978-953-307-326-2

Hard cover, 542 pages

**Publisher** InTech

**Published online** 09, September, 2011

**Published in print edition** September, 2011

This series is directed to diverse managerial professionals who are leading the transformation of individual domains by using expert information and domain knowledge to drive decision support systems (DSSs). The series offers a broad range of subjects addressed in specific areas such as health care, business management, banking, agriculture, environmental improvement, natural resource and spatial management, aviation administration, and hybrid applications of information technology aimed to interdisciplinary issues. This book series is composed of three volumes: Volume 1 consists of general concepts and methodology of DSSs; Volume 2 consists of applications of DSSs in the biomedical domain; Volume 3 consists of hybrid applications of DSSs in multidisciplinary domains. The book is shaped upon decision support strategies in the new infrastructure that assists the readers in full use of the creative technology to manipulate input data and to transform information into useful decisions for decision makers.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hugo López-Fernández, Florentino Fdez-Riverola, Miguel Reboiro-Jato, Daniel Glez-Peña and José R. Méndez (2011). Using CBR as Design Methodology for Developing Adaptable Decision Support Systems, Efficient Decision Support Systems - Practice and Challenges From Current to Future, Prof. Chiang Jao (Ed.), ISBN: 978-953-307-326-2, InTech, Available from: <http://www.intechopen.com/books/efficient-decision-support-systems-practice-and-challenges-from-current-to-future/using-cbr-as-design-methodology-for-developing-adaptable-decision-support-systems>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen