International Environmental Modelling and Software Society (iEMSs)
2010 International Congress on Environmental Modelling and Software
Modelling for Environment's Sake, Fifth Biennial Meeting, Ottawa, Canada
David A. Swayne, Wanhong Yang, A. A. Voinov, A. Rizzoli, T. Filatova (Eds.)
http://www.iemss.org/iemss2010/index.php?n=Main.Proceedings

Providing Intelligent Decision Support Systems with Flexible Data-Intensive Case-Based Reasoning

Beatriz Sevilla Villanueva [a] **and Miquel Sànchez-Marrè**[a]

[a]*Knowledge Engineering Machine Learning Group, Universitat Politècnica de Catalunya - BarcelonaTech, Barcelona, Spain (bsevilla@lsi.upc.edu, miquel@lsi.upc.edu)*

Abstract: In this paper we present a flexible CBR shell for Data-Intensive Case-Based Reasoning Systems which is fully integrated in an Intelligent Data Analysis Tool entitled GESCONDA. The main subgoal of the developed tool is to create a CBR Shell where no fixed domain exists and where letting the expert/user creates (models) his/her own domain. From an abstract point of view, the definition of the CBR can be seen as a methodology composed by four phases and each phase offers different ways to be solved. Then, since the CBR shell is integrated in GESCONDA, it inherits all its functionalities which cover the whole knowledge discovery and data mining process and also, CBR can complement its phases with this functionality. As a result, GESCONDA becomes an intelligent decision support tool which encompasses a number of advantages including domain independence, incremental learning, platform independence and generality.

Keywords: Instance-Based Reasoning Shell; Intelligent Data Analysis; Data Intensive Case-Based Reasoning; Intelligent Decision Support System

1 INTRODUCTION

1.1 Experiential knowledge in Environmental Processes Management and Operation

Management of environmental processes is a very difficult task, due to the complexity of the features involved in those systems, such as biological, chemical, physical, electrical, or ecological ones (detailed in Sànchez-Marrè et al. [2008]). Former mathematical control models have been used in their supervision and management, but some limitations of these approaches have been outlined. The main drawbacks are the lack of management of qualitative information, and the difficulty of using the expert and/or experiential knowledge about the process, which human experts usually obtain during several years of process operation.

The high quantity of information and implicit knowledge patterns contained in large databases coming from the monitoring of any dynamical environmental process is remarkable. Historical data collected about meteorological phenomena in a certain area, or about the performance of a wastewater treatment plant, or about characterizing environmental emergencies, or about geomorphological description of seismic activity are some examples.

Thus, most environmental systems can only be managed and supervised by experts using their own experience in the resolution of similar situations. These experts are not always accessible when dealing with risk situations, and it is crucial to record each new experience to learn about the process, while reusing this specific knowledge in the future. This is the reason why many artificial intelligence (AI) techniques have been used in recent past years trying to solve environmental processes management problem. Among those techniques, Case-Based Reasoning (CBR) has emerged as a very promising one. In environmental domains, exceptional situations commonly happen but there are not big amounts of data about these situations. Therefore, CBR could be more appropriated because it does not generalize the data but pretends to use specific cases in a particular domain to solve new cases. This analogical reasoning technique shows several advantages, like building solutions not from scratch, as other AI techniques, such as expert systems,

etc., and that the system is getting more reliable to solve problems as it is always learning new experiences (incremental learning).

Therefore, reliable Intelligent Environmental Decision Support Systems (IEDSS) should integrate this experiential knowledge of environmental processes gained through years of operation and management. Thus, Case-Based Reasoning functionality is a key component to be integrated in the building of reliable IEDSS.

1.2 Case-Based Reasoning

In this paper, a flexible and adaptable CBR shell integrated in a previous Knowledge Discovery and Data Mining tool is presented. This CBR shell is general enough to be parameterized and adjusted for the needs of whatever environmental process involved. From this goal two subobjectives are derived. The first is the design of a CBR shell being as flexible as possible in two senses. First, being extensible with new algorithms and highly customizable by the end-user. The second is the integration in GESCONDA, allowing the simultaneous use of CBR and GESCONDA functionality.

"*A case-based reasoner solves new problems by adapting solutions that were used to solve old problems*" by Riesbeck and Schank [1989]. According to Mántaras et al. [2006] and Aamodt and Plaza [1994], case-based reasoning, a reasoning paradigm and computational problem solving method, solves new problems by adapting previously successful solutions to similar experienced problems. CBR is gaining attention, since it does not require an explicit domain model and elicitation becomes a task of gathering historical cases, as described in Watson and Marir [1994].

The CBR formalization is summarized in a basic CBR system reasoning cycle, proposed by Aamodt and Plaza [1994], called the "4 RE's". This cycle has been the model for all the CBR systems that have been developed. The cycle is composed by the following four steps: **RE**trieve, **RE**use, **RE**vise and **RE**tain. Following the 4Re's cycle, the process starts with a new problem. The system *retrieves* the most similar previous situations, then the system *reuses* their solutions adapting those to the new problem. Once the proposed solution is applied, it can be evaluated (*revise*) and the system learns whether the proposed solution was appropriated to the new problem *retaining* this situation. In Case-based reasoning a case normally indicates a problem situation, or a previously experienced situation. Cases are adapted and learnt to solve future scenarios by simulating analogical reasoning, as described in Althoff [1999]. The great interest is drawn by the necessity of modeling human reasoning in Artificial Intelligence systems and particularly in Intelligent Decision Support Systems (IDSS), even in real-time IDSS.

Besides, most existing Case-Based Reasoning implementations focus on an specific domain, or at least those that want to cope with all CBR cycle. So, our purpose is to create an independent domain CBR Shell where the expert/user adapts her/his own domain. Hence, for a particular application domain, an intelligent decision support system is instantiated and the user provides a set of previously solved cases and configures the system in order to optimize his/her current domain management.

So far CBR has been introduced, the final subobjective is to integrate this flexible shell in a Knowledge Discovery in Databases (KDD) tool, called GESCONDA. This flexibility allows the tuning of the CBR system to get a better accuracy and reliability when solving management and operation tasks for each particular environmental process management. Additionally, it benefits from the fact that it is integrated in a KDD tool, and some preprocessing techniques like some variable transformations and the variable relevance determination by means of some feature weighting techniques, can be also applied to improve the results of the IDSS.

In Sànchez-Marrè et al. [2004] is described that GESCONDA was conceived to deal with environmental datasets. Therefore, the integration of the Case-Based Reasoning skills into GESCONDA tool is a first step to evolve GESCONDA from a data mining tool to a complete system for the development of IEDSS. Thus, not only the hidden knowledge models in the data can be discovered, but also they can be used for diagnosis and problem solving tasks. From a functional point of view, taking into account the type of problem that the IEDSS solves, an IEDSS aims to control or supervise a process in real-time (or almost real-time), facing similar situations. In general the end-user is responsible for accepting, refining or rejecting system solutions. This responsibility

can decrease, thereby increasing IEDSS confidence over time, as long as the system is facing situations that were successfully solved in the past (real validation).

The paper is organized as follows: next subsection introduces others related tools. The following section 2 briefly introduces GESCONDA. Section 3 presents the CBR shell: its main features and phases. In section 4 two databases are used to evaluate the whole system. Finally, section 5 points out some conclusions and future work.

1.3 Related Work

In the literature, it is possible to find a wide range of systems that provide help in the data mining stage, but not all provide full support for both the data mining and diagnosis, or problem solving tasks of an intelligent decision support. Among these systems, we emphasize the non-commercial suites. Probably the most known are Weka [2009] and RapidMider [2010]. Weka is a system that contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. In addition, it is offered as a API, so the functionality of Weka has been used for solving real-world data mining problems. RapidMiner (Yale) is a system for knowledge discovery and data mining. It fully integrates the Weka library. RapidMiner offers data Integration, analytical ETL, data analysis and reporting. Another less famous system is KEEL [2009] that contains a big collection of classical knowledge extraction algorithms, preprocessing techniques, computational intelligence based learning algorithms, including evolutionary rule learning algorithms, and hybrid models. It allows to perform a complete analysis of any learning model, including a statistical test module for comparison. In Alcalá-Fdez et al. [2009] there is a comparison of the most popular tools (by KD nuggets [2010]), where Keel is compared against WEKA, RapidMiner and others as KNIME [2010] and Tanagra [2004]. The comparison analysis shows that WEKA and Tanagra do not provide statistical tests and WEKA and RapidMiner have a lack in post-processing. Finally, none of them provides a mechanism to combine the results of the algorithms as GESCONDA does. Also, these systems allow both supervised and unsupervised learning algorithms, but just one attribute can be treated as a diagnosis, class or solution. However, GESCONDA introduces the multi-attribute as solution with CBR, and the authors are not aware of any previous IDSS that integrates a Case-Based Reasoning in a knowledge discovery tool.

There are some CBR shells available in the literature. Some of them are commercial products deployed in software companies (Remind, CBRExpress, Esteem etc.), while others are academic products developed by some academic research groups (Caspian, JColibri, IUCBR, AIAICBR, myCBR, etc.). Nevertheless, most of them do not allow implementing the whole basic CBR cycle in a very flexible way: retrieval, reuse, revise and retain. Since the research lines are focused on the retrieval step and the case base maintenance, not so many shells cope with all the phases, or at least the generic ones. jColibri2 [2009] is probably the strongest competitor as far as functionalities are concerned. Nowadays, JColibri2 is in fact one of the most popular shells, at least as an academic tool. Also, it is more used as an API and receives contributions from others entities extending its functionality. The case structure for all of the reviewed systems is built up with a list of attributes. Some of these systems support text as CBRExpress, ReMind and AIAICBR [2004]. But only JColibri2 and IUCBR [2009], along with our system, have extra attributes for assessing CBR algorithms. The last thing that can be remarked is that not all of the systems allow the solution to contain more than one attribute, such as AIAICBR or myCBR.

2 THE GESCONDA SYSTEM

GESCONDA is the name given to the Intelligent Data Analysis System with the aim of making Knowledge Discovery (KD) and intelligent data analysis. Although in the literature other KDD tools exist, none of them integrate, like GESCONDA , statistical and AI methods, the possibility of explicit management of the produced knowledge in Knowledge Bases (KB), mixed techniques that can cooperate among them to discover and extract the knowledge contained in data or dynamical data analysis in a single tool, allowing interaction among all methods as it is shown in Sànchez-Marrè et al. [2004]. Most of the results of the algorithms are included in the data as a new variable. Therefore, this new knowledge can be used afterwards in the implementation of reliable Intelligent Decision Support Systems.

On the basis of previous experiences, GESCONDA was designed as a multi-layer architecture of 4 levels connecting the user with the system or process, as it is introduced by Gibert et al. [2006]. The first level covers methods related to data filtering, including data cleaning through statistical tools, treatment and analysis of missing data and/or outliers; management and variable transformations; and graphical representations. In this pre-processing, different feature weighting approaches (supervised and unsupervised) are provided. The second level involves recommendation and meta-knowledge management. It supports the formal definition of problem goals, meta-knowledge of variables and examples and a recommender method that helps to select an appropriate method for data analysis is provided, parameter setting, and domain knowledge elicitation. The next is the knowledge discovery level. It includes several statistical, machine learning and data mining algorithms of classification and clustering, different bagging techniques, decision tree induction or rule based reasoning, among others, as well as some mixed techniques. The last level concerns the knowledge management that makes possible the integration of different knowledge patterns for a predictive task, or planning, or system supervision, as well as the validation of the knowledge patterns produced in the previous steps. User interaction is important in this phase, and the system supports it.

The last version of GESCONDA software has been developed in Java 5 following the Model-View-Controller (MVC) architecture pattern. In addition, it uses a LGPL library (jfreechart) for creating the graphics. About the implementation it is worth to mention that it is multithreading, that is being able to run different processes in parallel.

3 THE CBR SHELL

As it was mentioned before, the objective could be divided in two subobjectives: creating a CBR shell as flexible as possible and integrating it in GESCONDA. As GESCONDA is a KDD tool oriented to the data, the algorithms are based on the data without using complex structures or extra information about the domain (domain knowledge), so it is a Data Intensive CBR system.

We consider this shell flexible for two reasons. First, the design is conceived to allow the extension of different approximations in any phase of the cycle. Second, the existent techniques are highly customizable. To ease this customization, there is a configuration file where all the parameters that concern the selection of the methods to use in CBR can be set. Also, it can be done through the interface that it is integrated in GESCONDA. Figure 1 depicts the whole CBR cycle in a schematic way, and some of the parameters which can be customized at each step. Besides, it is indicated where the files could be required, either for a single cycle or when launching a battery of cases (see detailed description of the whole system in Sevilla [2009]).

In the following subsections the main features and methods of the CBR shell will be detailed: case structure, case library, and the four steps of the classical CBR cycle. To conclude, utility attribute and its use modes are introduced.

3.1 Case Structure

A case is a contextualized piece of knowledge that represents an experience. Normally, the case structure is divided into *Case Description* and *Case Solution*. Although, other attributes are interesting to improve the CBR performance such as the evaluation or the utility of the case.

The GESCONDA structure cannot be modified to guarantee the smooth running, in order to allow simultaneously using both the GESCONDA functionality and the CBR one. For this reason, the case structure has been adapted to the instance structure of GESCONDA what implies modifying the typical CBR model and reorganizing it to wrap the GESCONDA model.

From now on, the data structure called *instance* is shared, which is a vector of attribute-value pairs. These attributes can be continuous or discrete, and any of the discrete attributes can be selected as a class to be used in supervised learning in GESCONDA. Therefore, it was necessary to create a structure that allows to define which attributes belong to the case description or to the case solution part. This extra information can be introduced by the user manually through the interface or either be loaded by an xml file (see in figure1 *Descriptor* file). Besides the definition of description and solution of the new problem, it is possible to define other CBR attributes, such as *evaluation* and *utility*.

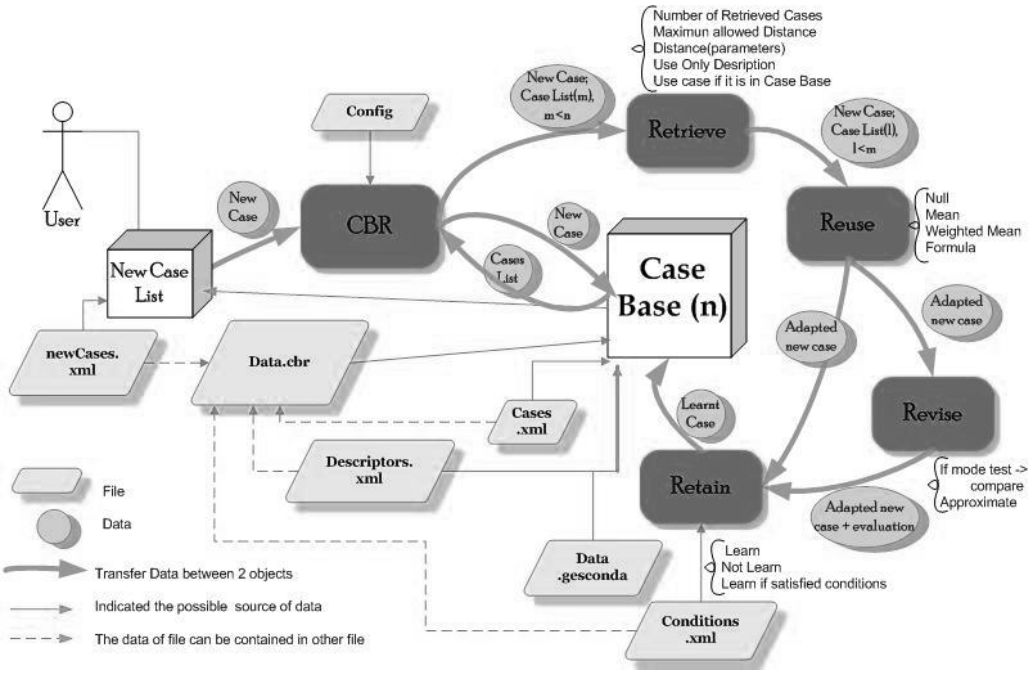

Figure 1: CBR architecture schema

3.2 Case Library

The system maintains the memory as it is done in GESCONDA which is a list or, in CBR terminology, a flat memory. The design and implementation allow an easy integration of other indexation schemes. There is a parameter to select the indexation type. But, at the moment, only flat indexation is implemented. Our next step is to integrate two types of indexation: Self Organizing Maps (SOM) and Hierarchical Structures. SOM described in I.K.Fodor [May, 2002], is especially useful to establish previously unknown linear and non linear relations among the cases. Hierarchical structures make the matching process and retrieval time more efficient as stated in Veloso and Carbonell [1993].

3.3 Retrieve

In general, retrieval phase involves finding the most similar case/s to a given new case. Usually this task is divided in two parts. First, a pre-selection of case set is made, depending on which indexation type is performed. Second, the retrieved cases are chosen from the last selection, filtering the cases which best match according to a similarity measure (nearest neighbour algorithm).

This software offers the possibility to define: the number of cases to retrieve, the maximum allowed distance, the weights of the attributes (manually or with GESCONDA implemented methods) and/or attribute standardization. The list of the retrieved cases is sorted by the similarity degree regarding the given case. The user can also select the similarity function (and its own parameters) to be used among the following:

L'Eixample. This distance is deeply described in Sànchez-Marrè et al. [1999]. It is an heterogeneous distance sensitive to weights. The most important parameter is a threshold, *alpha*, that defines the boundary for using quantitative or qualitative values for continuous attributes.

Minkowsky. The user can choose: to use weights, standardizing data and defining the variable power to specify the distance type. Power = 1 is the Manhattan distance and power=2 is the Euclidean distance – which is the most commonly used.

Cosinus This distance measures the degree of similarity along one space direction. It could be useful for some particular domains.

Canberra. Normalized distance often used for data scattered around the origin, introduced in Lance and Williams [1967].

Clark. Normalized distance sensitive to small changes close to the origin, explained in Eidenberger [2003].

3.4 Reuse

Reuse is inherently one of the most complex tasks in the CBR cycle since it requires a deep understanding of the represented situation in the retrieved case/s, in order to be able to modify their solutions. All this understanding is typically a domain dependent task. Thus, it is important to define generic reuse methods or methods which would be automatically able to capture the domain knowledge. In general, it is considered that the CBR inference is based on the principle as stated in Schank [1982]; Mántaras et al. [2006]; Lieber [2007]:

Similar problems have similar solutions

The adaptation of the retrieved case solutions focuses on the differences among the past and the new case and, also, what part of retrieved cases can be transferred to the new case. The selection of the cases to be used for adaptation among the retrieved cases can be guided by its *utility* attribute, by its *evaluation*, by its distance or either selected by the user. Once the retrieved cases are selected, the user can customize the adaptation algorithm to use it either for all solution attributes or customized for each one. In the literature, the reuse task is usually classified into the following types:

Null or Copy: the solution of the retrieved cases is directly transferred to the new case as its solution. Thus, the small differences are abstracted and they are considered as non relevant. So, this is a trivial type of reuse, but is widely used in many CBR systems since it is not domain dependent. In our CBR shell it is possible to apply this kind of method. When there is more than one retrieved case, a mean or mode value is computed.

Transformational Reuse: these methods transform the retrieved solution data into a new solution. The first approach developed in this CBR shell, domain independent, is the *Weighted Mean (Mode)*. The cases with high weight are more relevant (they influence more in the solution). This relevance could be chosen by the user. In spite it was conceived to be used with the *evaluation*, *utility* attribute and/or the distance to the new case. In order to be more generic, any attribute could be used as a parameter to weight the cases. In addition a formula module is provided where the formula for numerical attributes are defined by the user, thus allowing to create a suitable solution for adaptation. This formula model can be useful for environmental domains where there are interrelations among the attributes. For example, if two variables have the relation $v_{i1}/v_{i2} = v_{j1}/v_{j2}$ then – if the new case is j, and v_1 and v_2 are the solutions – then one of the variables could be solved by the formula.

Derivational Reuse: it reproduces the algorithms or methods that have been used to produce the retrieved solution. This adaptation is not based on the data but on how the solution of the retrieved case was assessed. Thus, this kind of reusing requires a more complex structure than the cases that this tool is dealing with. Also, the planning sequence that generated the original solution has to be stored in memory along with the solution. This approach, sometimes called *reinstantiation*, can only be used for cases that are well understood. This kind of reuse is not available in our CBR shell because it involves complex structures and domain knowledge.

3.5 Revise

This part of the cycle consists in the evaluation of the proposed solution to the new case. Therefore, it is indeed an important issue, because it is the opportunity to learn whether the previous task has performed correctly or not. It is supposed that this evaluation takes place after the solution is applied in a real environment, and then an expert evaluates how correct it has been. This phase is maybe the most difficult to deal with, because there is no automatic and generic methods implemented in the literature. Most of the CBR shells let the user do it himself or even avoid this part. Some systems simulate how the performance of the proposed case is, but it is needed domain knowledge to reproduce the real-world.

Two approaches are proposed and both include the evaluation attribute. Since this attribute it is optional, this phase could be avoided. The first looks at the evaluations of used cases to set up this new case and combine them. The second has to do with the testing mode (see 3.8) or the use of a case from the library; making in both cases possible to compare the estimated solution with

the real one and then evaluate it. In both approaches, the evaluation depends on the number of solution attributes and their types.

3.6 Retain

Learning by (own) experience is the last task. In each domain and application it is necessary to decide what, when and how a new experience has to be stored. Sometimes it is not worth to store it, i.e. the new case is almost equal to another case in the library. That is the reason for creating a functionality where the user can indicate waht the learning criteria are. The user has three options: to learn, not to learn o to learn if some specified conditions are satisfied. To the end user, a condition consists in three strings at any moment: name, operator or function, possible value (i.e, "similarity", "smaller", "0.8") and he/her must decide through the interface which conditions to use with their operators and values. Some of the conditions that are currently implemented are: based on the similarity to other cases or on evaluation or on utility and a recursive cluster elimination algorithm (RCE) described in Lim et al. [1991].

We present a flexible implementation where a developer user can add new "learning conditions". There is an xml file (see *conditions.xml* in figure1) where each condition is described and it will be loaded into the interface. Therefore, there are the condition definitions within the file: each condition has a name and a list of operators (function name) and, for each operator, a value list or a number. Also, the default operator and value to be used. In addition, the name of the *Condition*'s subclass where it is implemented is specified. This is a general formalization in order to not restrict what a condition could be. Hence, the retain algorithm only asks whether they are satisfied or not. Finally, the new case is retained if all the conditions are satisfied. This functionality will be reused to add deletion policies that will work offline or on demand. Some other algorithms will be integrated, such as those introduced in Smyth and McKenna [1999]; Orduña and Sànchez-Marrè [2008, 2009].

3.7 Utility

"The utility problem in artificial intelligence system occurs when knowledge learned in an attempt to improve a system's performance degrades performance instead", by Minton [March 1990]

"Utility" is a concept that tries to keep the usefulness of the case in the case base. This attribute may be useful, in many phases of CBR such as in *retrieve* and *adaptation,* to select more useful cases by giving them more relevance. Also, in the retain phase and case maintenance, by deleting those that are not so useful. However, always selecting the most useful cases creates the exploration vs. exploitation trade-off dilemma.

Our approximation expects to cover most of the domains by providing a flexible definition. Finally, some parameters are defined for all cases: number of times that a case has been retrieved, used for adaptation, used in a successful adaptation (that involves the *evaluation* attribute), its date of creation and its last use. The flexibility is due to the fact that the user can configure which are the parameters to be taken into account. In addition, in case that *utility* is defined as a frequency (as described in de la Rosa et al. [2007]), then a combination of numerical parameters is chosen, including the times that the CBR has been run.

3.8 Modes of use

First of all, the CBR shell can be run directly from the command line or either through the GESCONDA GUI. In both cases the performance is the same, and the configuration can be done through a *config* file (represented in figure 1). Also, in GUI mode the configuration could be set up through the interface.

It is possible to introduce a new case or a list of them (to launch a battery) to compute the CBR results or even use cases that belong to the case base, specifying whether they should be kept in the library. Also, if the cases come from the case base, there is the possibility to assess the evaluation according the similarities between the proposed solution and the real one. In addition, a *test* mode is included, where the solution attributes are duplicated with the proposed solutions. Thus, the observed and predicted solution can be compared and the following results can be calculated:

discrete solutions assess the percentage of success and continuous attributes the mean average error (mae), mean square error (msq), root means square deviation (rmsd), normalized root mean squared deviation or error (nrmsd), coefficient of variance (cv).

4 PRELIMINARY EVALUATION

The presented results come from two environmental domains and have been analyzed using CBR for predicting the solution attributes of each one of the instances of the databases.

The first one was the Abalone dataset. This database predicts the age of the abalone, measured as the number of rings, from some physical measurements. The abalones are a kind of mollusk, which in some world areas is considered as an endagered species. Therefore, a good estimation of the age of the abalones could have a great ecological impact. The Abalone Dataset [1994] from the UCI repository contains 4177 instances and 9 attributes (1 discrete, 8 continuous). In Clark et al. [1996] the authors propose to discretize the Rings attribute solution. In Waugh [1995], it is shown that the results fluctuate around 65%. In the experimentation, summarized in table 1, several CBR configurations using the developed CBR shell, like the similarity measures, the number of retrieved cases and the use of estimated weights using different methods were analyzed. As it can be seen in table 1, the accuracy results in the prediction of the age of abalones could range from 63.5% to an 82.6%, and for continuous results the p-values of the Student's $\mathcal{T}test$ are shown. Thus, it is very significant that the flexibility offered by the CBR shell could help the environmental scientist to find the most suitable configuration and to tune of the CBR system for making a better estimation of the age of the abalones.

Table 1: Results of applying different CBR configurations to Abalone dataset. The "Solution" column remarks whether the attribute Rings is continuous with "C" or either discrete by the number of classes. * values are assessed with the continuous values, later encapsulated into the 3 groups ($<$8; [9,10]; $>$11) and compared with the real values.

Distance	Num.Retrieves	Weigths	Solution	Success (%)	$\mathcal{T}test$	Other
Euclidean	3	-	Sex, Rings(C)	73.9	0.08	
Euclidean	3	-	Sex	77.4		
Euclidean	3	-	Rings(3)	83.7		
Euclidean	5	-	Rings(C)	63.5*	0.04	
L'Eixample	3	PROJ	Rings(3)	82		α =0.8
L'Eixample	5	PROJ	Rings(3)	80.8		α =1
L'Eixample	3	CVD	Rings(3)	82.6		α =0.8
Canberra	3	-	Rings(3)	80		
Cosinus	3	-	Rings(C)	64.5*	0.42	
Cosinus	5	-	Rings(3)	79.3		

The second dataset comes from a real wastewater treatment plan and contains 302 instances, 19 attributes (2 discrete) and 1471 missing values. For all the tests, the data have been standardized. A wastewater treatment plant is an environmental system, which is normally very difficult to manage and supervise, because many features are interacting (chemical, biological, physical, electrical, etc). In this database, the goal was to make a reliable diagnosis and to propose an output value for the 3 main control variables: Waste Activated Sludge flow (WAS), Recirculation Activated Sludge Flow (RAS), and Average Aeration flow (Qmedia), given a set of descriptive variables of the process. In the experimentation, summarized in table 2, several CBR configurations using the developed CBR shell, like the similarity measures, the number of retrieved cases, the use of estimated weights using different methods, the missing treatment of data, and the different adaptation schemes were tested. As a results, different error rates have been obtained (see section 3.8). Therefore, it is clear that the flexibility offered by the CBR shell helps the environmental scientist to find the most suitable configuration and to tune of the CBR system for controlling and supervising the wastewater treatment plant in a more reliable way.

5 CONCLUSIONS AND FUTURE WORK

The development of a flexible CBR shell for Data-Intensive Case-Based Reasoning Systems has been presented as well as its integration within a KDD Tool (GESCONDA). Additionally, the proposed tool is able to be adapted to each target domain in an easy way and highly customized. The integration in GESCONDA benefits the CBR performance because it can use its functionality.

Table 2: Results of applying different CBR configurations to Wastewater dataset. Missing: ignore (ignores every instances that have any missing), mean (replace missing data by the mean/mode). The error rates are between [0..1], being the lower the better.

Solution Attributes	Parameters	Distance	mae	mse	cv
WAS	num.retrieves:10	Euclidean	0,04721	0,01452	0,00539
RAS	missing:mean	no weights	0,02124	0,00689	0,01721
Qmedia	adaptation:mean		0,09359	0,01894	0,00205
WAS	num.retrieves:5	Euclidean	0,04594	0,01404	0,00530
RAS	missing:mean	weights:ueb1	0,01694	0,00655	0,01678
Qmedia	adaptation:mean		0,10119	0,02094	0,00215
WAS	num.retrieves:5	Euclidean	0,04594	0,01404	0,00530
RAS	missing:mean	weights:ueb1	0,01694	0,00655	0,01678
Qmedia	adaptation:weightedmean		0,10119	0,02094	0,00215
Filtered Data: Deleted Attributes with missing values $>$ 50% & instances with missing values$>$80%					
WAS	num.retrieves:10	Euclidean	0.04244	0.01227	0.00575
RAS	missing:none	no weights	0.02138	0.00763	0.02047
Qmedia	adaptation:mean		0.09825	0.01887	0.00246
WAS	num.retrieves:10	L'Eixample	0.04558	0.01302	0.00534
RAS	missing:mean	$\alpha = 0.8$	0.02181	0.00734	0.01856
Qmedia	adaptation:weightedmean	no weights	0.08726	0.01635	0.00199
WAS	num.retrieves:5	Euclidean	0.03583	0.00935	0.00623
RAS	missing:mean	weights:ueb1	0.02088	0.00902	0.02400
Qmedia	adaptation:mean		0.10037	0.02113	0.00298
WAS	num.retrieves:5	Euclidean	0.04564	0.01308	0.00556
RAS	missing:mean	no weights	0.02285	0.00892	0.02051
Qmedia	adaptation:weightedmean		0.09056	0.01705	0.00204
WAS	num.retrieves:10	Euclidean	0.04495	0.01307	0.00534
RAS	missing:mean	weights:ueb1	0.01909	0.00727	0.01847
Qmedia	adaptation:weightedmean		0.09246	0.01811	0.00209

Most of the algorithms in GESCONDA store the results as a new attribute, improving CBR power by using this new attribute. For instance, having a classification algorithm which has just tagged all cases in *n* classes, this new *class* attribute could be helpful for measuring the distance between similar individuals.

Eventually, it is worth to mention that the CBR shell lets the users to model whatever domain, including high dimensional domains and long datasets. Especially mentionable is the possibility to use all the available capabilities within the Intelligent Data Analysis tool (i.e., GESCONDA software) making easier the task of data preparing, data filtering, feature weighting, data visualization, etc., which becomes a not commonly found feature in most of available CBR shells in the literature.

Summarizing, the integration of this flexible CBR shell into GESCONDA provides it with actual reasoning abilities. It evolves GESCONDA to a useful Intelligent Decision Support System. Since GESCONDA was conceived to support environmental databases and CBR works with the same data model, the result is a system suitable for environmental domains.

At present GESCONDA is integrating a rule based reasoning that could be used by CBR adaptation to generate the new solution. Also, other internal features of the CBR shell will be improved. For instance, the case library structure and the introduction of the deletion policies. Other skills to make possible the monitoring of the system and the capability of generating some graphical charts about several important parameters of the system (case library size, retrieval time, etc.) will be available in next releases of the tool.

REFERENCES

Aamodt, A. and E. Plaza. Case-based reasoning: Foundational issues, methodological varations and system approaches. *AI Communications. IOS Press*, 7(1):39–59, 1994.

Abalone Dataset. archive.ics.uci.edu/ml/datasets/Abalone, 1994.

AIAICBR. *www.aiai.ed.ac.uk/project/cbr/CBRDistrib*, 2004.

Alcalá-Fdez, J., L. Sánchez, S. García, M. del Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernández, and F. Herrera. Keel: A software tool to assess evolutionary

algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009.

Althoff, K.-D. *Case Base Reasoning Research and Development*. Springer Verlag, 1999.

Clark, D., Z. Schreter, and A. Adams. A quantitative comparison of dystal and backpropagation. *Australian Conference on Neural Networks (ACNN'96)*, 1996.

de la Rosa, T., A. G. Olaya, and D. Borrajo. Using cases utility for heuristic planning improvement. In *Case-Based Reasoning Research and Development*, volume 4626/2007, pages 137–148. Springer Berlin / Heidelberg, 2007.

Eidenberger, H. Distance measures for mpeg-7-based retrieval. *5th ACM SIGMM international workshop on Multimedia information retrieval*, 2003.

Gibert, K., M. Sànchez-Marrè, and I. Rodríguez-Roda. Gesconda: An intelligent data analysis system for knowledge discovery and management in environmental data bases. *Environmental Modelling & Software 21(1):116-121*, 2006.

I.K.Fodor. A survey of dimension reduction techniques. *UCRL*, May, 2002.

IUCBR. *http://www.cs.indiana.edu/∼sbogaert/CBR/*, 2009.

jColibri2. *gaia.fdi.ucm.es/projects/jcolibri*, 2009.

KD nuggets. *www.kdnuggets.com/software*, 2010.

KEEL. *www.keel.es/*, 2009.

KNIME. *www.knime.org*, 2010.

Lance, G. N. and W. T. Williams. Mixed-data classificatory programs i - agglomerative systems. *Australian Computer Journal*, 1(1):15–20, 1967.

Lieber, J. Application of the revision theory to adaptation in case-based reasoning: the conservative adaptation. *7th International Conference on Case-Based Reasoning - ICCBR'07*, 2007.

Lim, J., H. Lui, and A. Tan. Inside: a connectionist case-based diagnostic expert system that learns incrementally. *IEEE International Joint Conference on Neural Networks*, 1991.

Mántaras, R. L., D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M.-L. Maher, M. Cox, K. Forbus, M. Keane, and I. Watson. Retrieval, reuse, revision, and retention in cbr. *The Knowledge Engineering Review*, 20(3):215–240, 2006.

Minton. Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence, 42*, March 1990.

Orduña, F. and M. Sànchez-Marrè. Case base maintenance: Terms and directions. *Report*, 2008.

Orduña, F. and M. Sànchez-Marrè. The dynamic adaptative case-based library for continuous domains. *CCIA*, pages 157–166, 2009.

RapidMider. *http://rapid-i.com/*, 2010.

Riesbeck, C. K. and R. C. Schank. *Inside case-based reasoning*. Lawrence Erlbaum Associates, Pubs., Hillsdale, N.J., 1989.

Sànchez-Marrè, M., U. Cortés, I. R. Roda, and M. Poch. Sustainable case learning for continuous domains. In *Environmental Modelling and Software Volume 14, Issue 5*, pages 349–357, 1999.

Sànchez-Marrè, M., K. Gibert, and I. Rodríguez-Roda. Gesconda: A tool for knowledge discovery and data mining in environmental databases. In *e-Environment: Progress and Challenge. Research on Computing Science, Vol. 11, CIC, Mexico*, pages 348–364, 2004.

Sànchez-Marrè, M., K. Gibert, R. S. Sojda, J. P. Steyer, P. Struss, I. Rodríguez-Roda, J. Comas, V. Brilhante, and E. A. Roehl. *Intelligent Environmental Decisions Support Systems*, pages 119–144. Elsevier, 2008.

Schank, R. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, 1982.

Sevilla, B. Design and development of a case-based reasoning in an intelligent data analysis tool. Master's thesis, Universitat Plitècnica de Catalunya. BarcelonaTech, 2009.

Smyth, B. and E. McKenna. Building compact competent case-bases. In *Proceedings of the Third International Conference on Case-Based Reasoning*, pages 329–342. Springer, 1999.

Tanagra. *eric.univ-lyon2.fr/ricco/tanagra/en/tanagra.html*, 2004.

Veloso, M. M. and J. G. Carbonell. Derivational analogy in prodigy: automating case acquisition, storage and utilization. pages 249–278, 1993.

Watson, I. and F. Marir. Case-based reasoning: A review. *AI-CBR, Dept.of Computer Science, University of Auckland, New Zealand*, 1994.

Waugh, S. *Extending and benchmarking Cascade-Correlation*. PhD thesis, Computer Science Department, University of Tasmania., 1995.

Weka. *www.cs.waikato.ac.nz/ml/weka*, 2009.