COMPARATIVE STUDY OF CASE BASED REASONING SOFTWARE

Passent ElKafrawy*, Rania. A. Mohamed**
*Faculty of Science, Menofia University, Egypt
** Faculty Computer Science, Modern University for Technology &Information, Cairo, Egypt

ABSTRACT

Case-Based Reasoning (CBR) is a problem-solving paradigm that solves a new problem by remembering a previous similar situation and by reusing the information and knowledge of that situation. A Case-Based Reasoning (CBR) tool is software that can be used to develop several applications that require case-based reasoning methodology. CBR is based on the intuition that information gained from previous experiences (cases or instances) can be an important tool to provide solutions for, and enhance processes related to, the problem at hand. Therefore, it helps improve outcomes and saving valuable resources. However, large volumes of information can make it a complex task to gain useful insight from historic datasets. CBR is an effective tool to gain useful information in such circumstances and is one of the most successfully applied Artificial Intelligence technologies in recent years. This paper gives background information about CBR software and introduces the most used CBR tools (CBR Shell, FreeCBR, jCOLIBRI, myCBR and eXiTCBR). Then it introduces a comparative analysis study based on some determined factors that affect the CBR software including noisy data or missing values in the cases. It also compares between them from the? Finally, an evaluation of the retrieving phase of each software is introduced.

KEYWORDS: *- Artificial Intelligence, Case-Based Reasoning.*

I. INTRODUCTION

Case-based Reasoning is an emerging field in Artificial intelligence. It is mostly used in problem solving in the artificial intelligence applications. Case-based reasoning is an approach which utilizes the experience gained from solving past problems [1]. This approach maintains all information of past-solved problems, where this experience is stored as a case. The collection of all these past cases is stored in the form of case-base.

There are various factors which define the efficiency of this approach [2]. The major factor is the number of past experiences stored in a case base. The new problem should be identified in term of the experience of the problems faced before. The new upcoming problem is considered as a new case. The strategy of finding a similar case for the new problem under investigation stored in the case base is another major factor of defining the efficiency of the case-based reasoning approach. The evaluation of the selected case and indexing of the suggested case for future use are another factors that define the performance of case-based reasoning system.

Case-based reasoning has many advantages over other reasoning approaches such as rule based reasoning [5]. This reasoning approach bears a resemblance to human reasoning. It provides the facility of taking the decision such as human beings take decision in real time. Case-based reasoning is a machine learning mechanism as the solutions of precedent problems faced are stored in the case base. This approach learns from both success & failure of solutions of the previous problems. These past experiences are being reused for solving the coming problems. The process of knowledge acquisition is easily handled in this approach. But in the case of other reasoning approaches, the knowledge acquisition process is not so simple & also costly. The other major advantages of this approaches over the other reasoning approaches is the worth of the solution. In the revise phase of case-based reasoning approach, the proposed solution is revised according constraints of the problem. Then the proposed solution is repaired according to constraints. It is also modified for fulfilling the constraints of the problem. This phase of case-based reasoning boosts the excellence of the solutions & extends the effectiveness of this approach. The errors of the previous solutions do not propagate in the future of problem's solutions [5]. It can also be applied in those domains where the information about problems is incomplete & insufficient for finding the adequate rules or algorithms to solve them.

After this introduction, a theoretical background is illustrated in section 2. In section 3

II. THEORETICAL BACKGROUND

Case-Based Reasoning (CBR) is a problem solving paradigm that solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation [1]. More specifically, CBR uses a database of problems to resolve new problems. The database can be built through the knowledge Engineering (KE) process or it can be collected from previous cases.

In a problem-solving system, each case would describe a problem and a solution to that problem. The reasoning engine solves new problems by adapting relevant cases from the library [3]. Moreover, CBR can learn from previous experiences. When a problem is solved, the case-based reasoning can add the problem description and the solution to the case library. The new case that in general represented as a pair <problem, solution> becomes immediately available and can be considered as a new piece of knowledge.

According to Doyle et al. [4], Case-Based Reasoning is different from other Artificial Intelligence approaches in the following ways:

Traditional AI approaches rely on general knowledge of a problem domain and tend to solve problems on a first-principle while CBR systems solve new problems by utilizing specific knowledge of past experiences.

CBR supports incremental, sustained learning. After CBR solves a problem, it will make the problem available for future problems.

The CBR Cycle can be represented by a schematic cycle, as shown in Figure 1. First phase is the retrieve phase, which identifies features via noticing the feature values of a case, initially match a list of possible candidates and select the best match from the cases[7].

Second phase is the reuse phase, where the difference between the new and the old case is determined by copying from the old case and adapting by transforming or reusing the old solution.

The third phase is the revise phase, if the solution from the last phase is incorrect, then this solution must be evaluated in a real environment setting and the errors/flaws of the solution must be found if the solution was evaluated badly.

Finally, the Retain phase which incorporates the lesson learned from the problem-solving experience into the existing knowledge by extracting or indexing. By extracting we mean if the problem was solved using an old case, the system can build a new case or generalize an old case. By indexing we mean via deciding what types of indexes can be used in the future by integrating and modifying the indexing of existing cases after the experience.

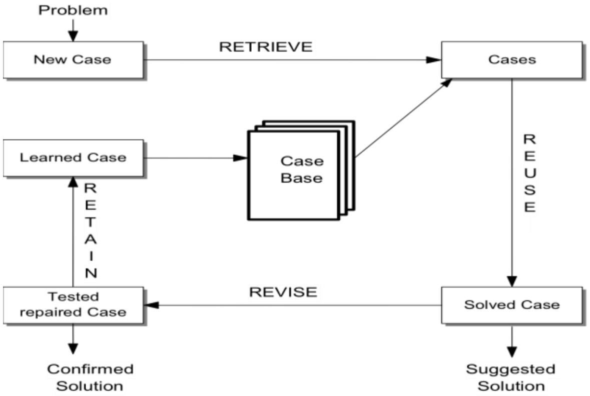


Figure 1: Case-based Reasoning

There are three main types of CBR that differ significantly from one another concerning case representation and reasoning. The first one is called Structural in which a common structured vocabulary is developed, i.e. ontology. The second is textual in such way cases are represented as free text, i.e. strings. The third is the Conversational CBR in which a case is represented through a list of questions that vary from one case to another; knowledge is contained in customer / agent conversations [5].

During the past twenty years, many CBR applications have been developed, ranging from prototypical applications built in research labs to large-scale fielded applications developed by commercial companies[6].

The common application areas of CBR include help-desk and customer service, recommender systems in electronic commerce, knowledge and experience management, medical applications and applications in image processing, applications in law, technical diagnosis, design, planning and applications in the computer games and music domain [6].

CBR shells are kinds of application generators with graphical user interface. Nonprogrammer users can use them but the extension or integration of new components in these tools are not possible.

There is a clear difference between a CBR application and a CBR shell. A CBR application is a direct implementation of CBR methodology to a specific domain problem in order to solve this problem. On the other hand, a CBR shell is an application that enables developers to develop a CBR application.

III. CBR SOFTWARE

This section introduces a brief description of each software used in this paper

Table 1: CBR software

CBR Shell	Author
CBR Shell	AIAI, Stuart Aitken
FreeCBR	Lars Johanson
jCOLIBRI	University Complutense Madrid, GAIA group
myCBR	German Research Center for Artificial Intelligence
eXiTCBR	University of Girona

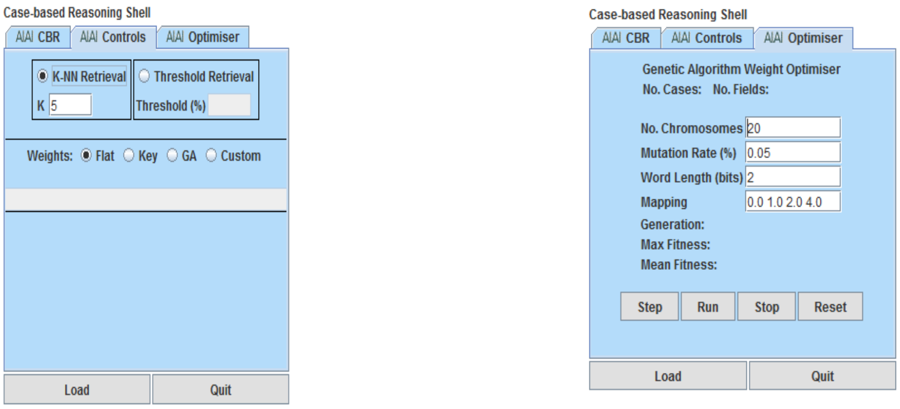
3.1CBR Shell

The AIAI CBR [8] Shell is a generic tool for case-based reasoning. The tool performs classification based on case comparison. The parameters of the algorithm can be varied: the number of nearest neighbors considered can be specified, the weights can be set manually, or the weights can be optimized by genetic algorithm. The accuracy of the algorithm is measured by a leave-one-out evaluation.

The data must be in comma-delimited form, where a newline delimits a case. The first line of the case base must contain the name of the key file, the second states the goal field. The key file defines the type of matching that is done on each field in each case. The matching types include:

- Num-numerical comparison by evaluating the ratio of 2 numbers
- Stringexact-string comparison (equality test)
- Trigram-comparison of strings/sentences/paragraphs by trigram matching

Figure 2 shows a sample screen shots of the application while testing a sample case base.



● K-NN Retrieval ○ Threshold Retrieval

K 5

Threshold (%)

Weights: ● Flat ○ Key ○ GA ○ Custom

Genetic Algorithm Weight Optimiser

No. Cases: No. Fields:

No. Chromosomes 20

Mutation Rate (%) 0.05

Word Length (bits) 2

Mapping 0.0 1.0 2.0 4.0

Generation:

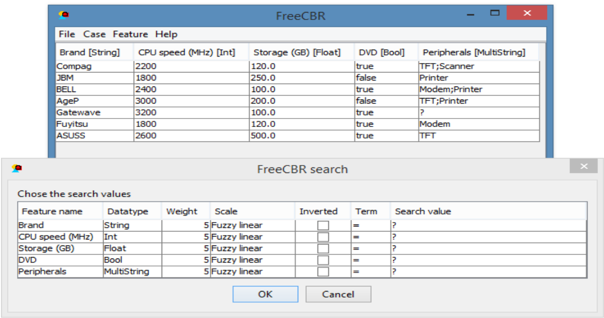
Max Fitness:

Mean Fitness:

Figure 2: CBR Shell GUI interface

3.2FreeCBR

FreeCBR[9] is a free open source Java implementation of a Case Based Reasoning tool. Cases are stored as text cases; each case is a set of features. Each case consists of a predefined set of features. It finds the closest match among cases in a case set. The closest match is calculated using weighted Euclidian distance, Normal Distance algorithm, and Logarithmic Distance algorithm. It only supports selection and retrieval phases.



Brand [String]	CPU speed (MHz) [Int]	Storage (GB) [Float]	DVD [Bool]	Peripherals [MultiString]
Compag	2200	120.0	true	TFT;Scanner
JBM	1800	250.0	false	Printer
BELL	2400	100.0	true	Modem;Printer
AgeP	3000	200.0	false	TFT;Printer
Gatewave	3200	100.0	true	?
Fuyitsu	1800	120.0	true	Modem
ASUSS	2600	500.0	true	TFT

Chose the search values

Feature name	Datatype	Weight	Scale	Inverted	Term	Search value
Brand	String	5	Fuzzy linear	□	=	?
CPU speed (MHz)	Int	5	Fuzzy linear	□	=	?
Storage (GB)	Float	5	Fuzzy linear	□	=	?
DVD	Bool	5	Fuzzy linear	□	=	?
Peripherals	MultiString	5	Fuzzy linear	□	=	?

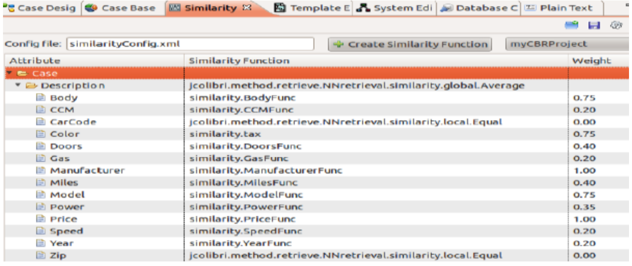
Figure 3: FreeCBR

Figure 4: jCOLIBRI GUI

3.3jcolibri

jCOLIBRI [10] is a framework for developing various CBR applications. It is Java-based and uses JavaBeans technology for case representation and automatic generation of user interfaces.

jCOLIBRI supports full CBR cycle. At Retrieve stage, the nearest N cases are retrieved and there are 5 retrieval strategies, 7 selection methods and over 30 kinds of similarity functions (SF) in the spheres of text formatting and ontology. At the Refusal stage, several methods for adaptation are available (direct proportion) and also in ontology. At Revise stage, methods for revision of cases are realized, as well methods for new indexes (IDs) generation and methods for decision making (preference elicitation). At Retain stage, there are methods for query retaining as a new case. The maintenance algorithms such RENN, BBNR, etc. are also supported. jCOLIBRI allows retrieval form clustered and indexed case bases and submits program interfaces (connectors) to access text and XML files, as well standard and DL (descriptive logic) data bases. These interfaces can be used for diagnostic systems databases access. All CBR cases can be represented graphically. There are lots of CBR applications, developed on jCOLIBRI base: additional shells (abstract levels) for distributed CBR systems, statistical CBR systems, multiagent supervisor systems [10, 11,12], systems for text files classification, and a lot of CBR recommender systems intended to trip, car type or restaurant choice, and other of discrete-event type. Figure 4 shows a sample screen shot of jCOLIBRI

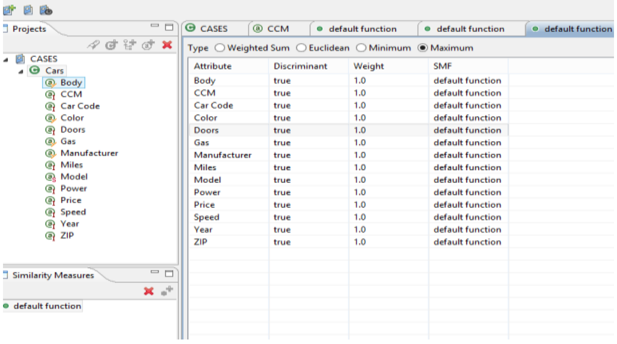


Config file: similarityConfig.xml

Attribute	Similarity Function	Weight
Case		
Description	jcolibri.method.retrieve.NNretrieval.similarity.global.Average	
Body	similarity.BodyFunc	0.75
CCM	similarity.CCMFunc	0.20
CarCode	jcolibri.method.retrieve.NNretrieval.similarity.local.Equal	0.00
Color	similarity.tax	0.75
Doors	similarity.DoorsFunc	0.40
Gas	similarity.GasFunc	0.20
Manufacturer	similarity.ManufacturerFunc	1.00
Miles	similarity.MilesFunc	0.40
Model	similarity.ModelFunc	0.75
Power	similarity.PowerFunc	0.35
Price	similarity.PriceFunc	1.00
Speed	similarity.SpeedFunc	0.20
Year	similarity.YearFunc	0.20
Zip	jcolibri.method.retrieve.NNretrieval.similarity.local.Equal	0.00

3.4myCBR

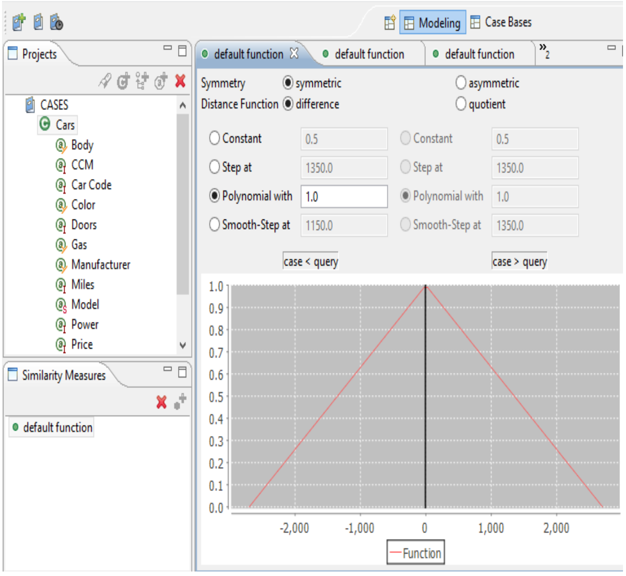
myCBR [13] is an open-source similarity-based retrieval tool and software development kit (SDK). The framework my CBR supports description of cases with various attributes: numeric, character and string, logical, class type, etc. The templates of the cases are generated as classes or subclasses with a number of attributes, called slots (Figure 6).



Type ○ Weighted Sum ○ Euclidean ○ Minimum ● Maximum

Attribute	Discriminant	Weight	SMF
Body	true	1.0	default function
CCM	true	1.0	default function
Car Code	true	1.0	default function
Color	true	1.0	default function
Doors	true	1.0	default function
Gas	true	1.0	default function
Manufacturer	true	1.0	default function
Miles	true	1.0	default function
Model	true	1.0	default function
Power	true	1.0	default function
Price	true	1.0	default function
Speed	true	1.0	default function
Year	true	1.0	default function
ZIP	true	1.0	default function

Figure6: Editing the local similarity function viaGUI



Symmetry ● symmetric ○ asymmetric
Distance Function ● difference ○ quotient

○ Constant 0.5 ○ Constant 0.5
○ Step at 1350.0 ○ Step at 1350.0
● Polynomial with 1.0 ● Polynomial with 1.0
○ Smooth-Step at 1150.0 ○ Smooth-Step at 1350.0

Fig.5. myCBR case base

.The CBR Cases are objects of the class described by its attributes. Each attribute can participate in the class with its value and weight that determine the significance of the attribute in relation to others. Attributes with weight of zero (0) are not considered when searching the case-base DB.

Usually, case decisions have attributes with zero weight. In myCBR are given the opportunity to edit the similarity functions (SF) on class level (global SF) and on an attribute level (local SF). At the class level the SF are: weighted sum, Euclidean difference, maximum or minimum. On attribute level, the SF can be modified through the GUI, as shown in the middle of Figure 5 and they can be symmetrical, asymmetrical, step-type or smooth step-type, linear or polynomial.

In myCBR the case and their attributes can be created manually or automatically. The automatic generation of attributes (slots) is done during the import procedure of the Comma Separated Value (CSV) file. Then to each column name from the CSV file is assigned an attribute with the same name. To each row of the file the new case (instance of the class) is created in the case-base DB.

3.5 exitcbr

eXiTCBR [14] is a case-based reasoning tool developed at the eXiT research group of the University of Girona. It goes beyond pure CBR prototyping and aims to support experimentation.

The eXiTCBR framework was designed to bring together CBR methods currently used in medical applications and data mining and visualization techniques that can be plugged into it. eXiTCBR also facilitates the incorporation of new techniques, if required.

eXiTCBR architecture follows a modular approach based on the different phases in a CBR system. Each CBR step is implemented as generic class. When a new method is required but not provided in the system, it can be assembled as a particular instance of a generic class. When other techniques need to be integrated or hybridized, the corresponding executable codes should also be included.

Input file Requirements:[15]

- First row: description of the attributes
- Second row: short name of the attributes, in a single word
- Third row: attribute type:

0: discrete (as Nause, LumbarPain, Urine, Micturition, Burning)

1: numeric (as Temp)

2: text

-1: do not take into account (as identifier, or the classes).

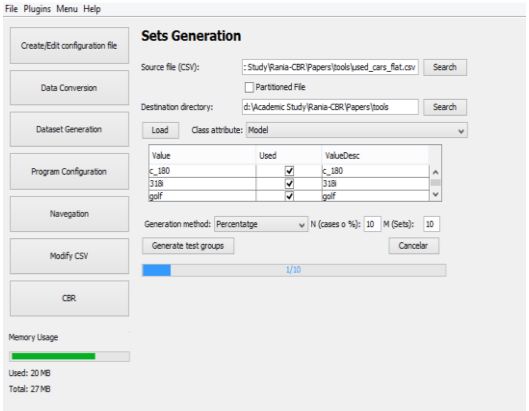
-Fourth row: attribute weight

- First Column: case identifier. There should be a different identifier for all of the cases
- Class attribute: it should be numeric (0-positive, 1-negative, or the other way around). In the example, Inflammatory is a diagnosis (class attribute), as well as nephritis.
- No empty lines at the end of the file
- Decimal numbers expressed with dots (39.0). Do not use dots for thousands.

-Current eXITCBR version provides support for the first two steps.

In selection the retrieved cases, the selection methods select the best cases from with the solution of a new case is provided at the reuse phase. The selection methods available are the following:

- Select1K. The best / most similar case will only be used.
- Select NK. The N best similar cases will be used.
- Select Threshold. The cases close to the new case will be used. “Close” is modelled by a threshold that needs to be specified at the right box. Use this option as a first attempt to use the tool, indicating in the right box 0.7 this option illustrates in a nice way the plots that result after running the experiment.
- Null. No method is applied. All the cases are used.



Sets Generation

Source file (CSV): : Study\Rania-CBR\Papers\tools\used_cars_flat.csv

□ Partitioned File

Destination directory: d:\Academic Study\Rania-CBR\Papers\tools

Class attribute: Model

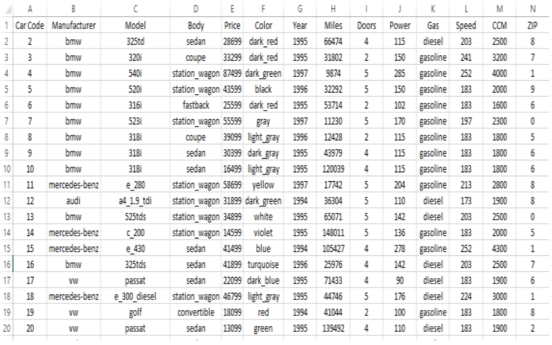
Value	Used	ValueDesc
c_180	☑	c_180
318i	☑	318i
golf	☑	golf

Generation method: Percentatge N (cases o %): 10 M (Sets): 10

Used: 20 MB

Total: 27 MB

Figure 7: eXiTCBR CBR



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Car Code	Manufacturer	Model	Body	Price	Color	Year	Miles	Doors	Power	Gas	Speed	CCM	ZIP
2	2	bmw	325td	sedan	28699	dark_red	1995	66474	4	115	diesel	203	2500	8
3	3	bmw	320i	coupe	33299	dark_red	1995	31802	2	150	gasoline	241	3200	7
4	4	bmw	540i	station_wagon	87499	dark_green	1997	9874	5	285	gasoline	252	4000	1
5	5	bmw	520i	station_wagon	43599	black	1996	32292	5	150	gasoline	183	2000	9
6	6	bmw	316i	fastback	25599	dark_red	1995	53714	2	102	gasoline	183	1600	6
7	7	bmw	523i	station_wagon	55599	gray	1997	11230	5	170	gasoline	197	2300	0
8	8	bmw	318i	coupe	39099	light_gray	1996	12428	2	115	gasoline	183	1800	5
9	9	bmw	318i	sedan	30399	dark_gray	1995	43979	4	115	gasoline	183	1800	6
10	10	bmw	318i	sedan	16499	light_gray	1995	120039	4	115	gasoline	183	1800	6
11	11	mercedes-benz	e_280	station_wagon	58699	yellow	1997	17742	5	204	gasoline	213	2800	8
12	12	audi	a4_1.9_tdi	station_wagon	31899	dark_green	1994	36304	5	110	diesel	173	1900	8
13	13	bmw	525tds	station_wagon	34899	white	1995	65071	5	142	diesel	203	2500	0
14	14	mercedes-benz	c_200	station_wagon	14599	violet	1995	148011	5	136	gasoline	183	2000	5
15	15	mercedes-benz	e_430	sedan	41499	blue	1994	105427	4	278	gasoline	252	4300	1
16	16	bmw	325tds	sedan	41899	turquoise	1996	25976	4	142	diesel	203	2500	7
17	17	vw	passat	sedan	22099	dark_blue	1995	71433	4	90	diesel	183	1900	6
18	18	mercedes-benz	e_300_diesel	station_wagon	46799	light_gray	1995	44746	5	176	diesel	224	3000	1
19	19	vw	golf	convertible	18099	red	1994	41044	2	100	gasoline	183	1800	8
20	20	vw	passat	sedan	13099	green	1995	139492	4	110	diesel	183	1900	2

Figure 8: Case base Snapshot

IV. COMPARATIVE STUDY

This section introduces a comparative study after testing and comparing the CBR applications mentioned previously in table 1 using the same case base.

The case base used for testing the previously mentioned software obtained from the UC Irvine Machine Learning Repository which contains details for 1000 cases for used cars [16].

There are a number of major concerns when studying case-based reasoning approach. These major concerns are listed below:

What is the structure of the cases?

- What are the selection strategies for finding similar case?
- How is the case being retrieved?
- How is the selected case being revised?
- How is the suggested case being stored in case base?
- How is the suggested case being indexed for faster access?
- How to deal with noisy data or missing values?

According to the previous mentioned points, a comparative study between the CBR software mentioned previously in table 2 in section 3 is done. Next paragraphs describe the effect of each factor to each CBR software respectively.

After applying the same query to all CBR software, the researcher has discovered the following: CBR Shell, very simple interface, the retrieval can use KNN or Threshold and weights can be specified manually, it uses genetic algorithm for optimization, no case revised and cases are stored in custom text files, no case indexing.

FreeCBR has a very simple GUI interface, find the "closest" match of the stored cases, the closest match is calculated using weighted Euclid distance. FreeCBR finds the closest match among cases in a case set. Each case consists of a predefined set of features. The features are defined by a name and a data type where the data type may be String, MultiString, Float, Int and Bool.

jColibri has a very simple and powerful GUI, it represents cases in a very simple way. jColibri allows retrieving cases using a SQL query and then it organizes cases after they load into memory and the case can be graphically presented.

There are a number of case retrieval algorithms applicable in case based reasoning. These algorithms are based on the similarity metric that allows resemblance between cases stored in case base. The nearest neighbor retrieval algorithm & induction retrieval algorithms are two chief algorithms used in this process. Nearest-neighbor retrieval is a straightforward approach that computes the similarity between relevant cases found through indexing. The case is elected on worth of weighted computation of its feature. When the value of weighted calculation of its features is greater than other cases, then meticulous case is elected from the case base.

jCOLIBRI can be used as a basis for complex CBR applications development with full CBR R4 cycle, using various data bases.

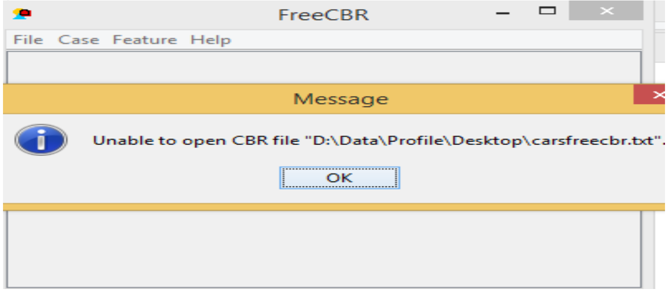
jCOLIBRI supports working with external Database and external sources.myCBR has a simple GUI. The cases are very simple. They support only Retrieve and Retain phases. During the Retrieve phase, all precedents are extracted. They are sorted by degree of similarity based on the chosen global SF. The Query to the case-base DB could be done on the basis of all or part of the attributes, describing the case.

myCBR does not work with external DB. It stores the cases in text file or in XML file. That's way it cannot support the case indexation and categorization. The case cannot be graphically presented in the GUI, but it is possible to present the distribution of values of a selected attribute for all cases in the database.

No interfaces to external systems and DB are available in myCBR. It is valid regarding the interfaces to real-time or diagnostic systems. On Retain phase, myCBR allows saving the Query as a new case, also to use an old case as a basis for new Query. MyCBR is entirely based on GUI, providing a ready-windows templates and forms for defining classes, attributes, SFs, queries to the case-base DB, visualization of found results and more.

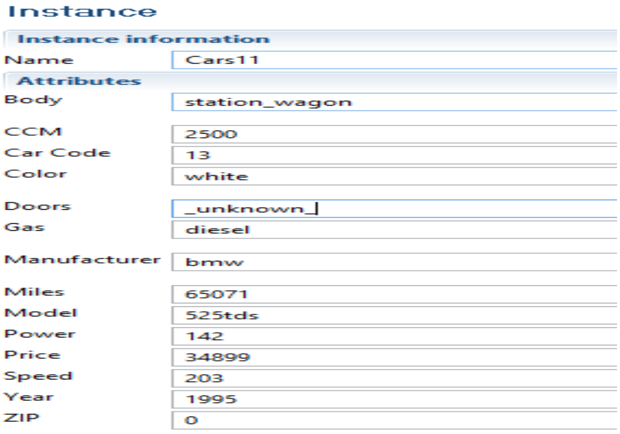
myCBR platform can be used for non-complex CBR applications development with partial CBR R4 cycle and with small number of cases in text file. For CBR application development, no time for programming is needed but it is needed only for case configuration. MyCBR is not suitable to be applied with large number of attributes with text solution, especially when they must be visually presented in one window. Table 2 summarizes the comparisons between the selected CBR software.

Concerning the missing data and unknown values in the case base software, CBR Shell, FreeCBR, and eXiTCBR can’t load the case base if modified or edited with removing values as shown in figure 9. myCBR handles this type of noise by adding two extra values (_unknown_ and _undefined_) to an attribute as shown in figure 10.



Unable to open CBR file "D:\Data\Profile\Desktop\carsfreecbr.txt".

Figure 9: FreeCBR and missing values



Instance

Instance information

Name	Cars11

Attributes

Body	station_wagon
CCM	2500
Car Code	13
Color	white
Doors	_unknown_
Gas	diesel
Manufacturer	bmw
Miles	65071
Model	525tds
Power	142
Price	34899
Speed	203
Year	1995
ZIP	0

Figure 10 :myCBR and missing values

As an addition to the comparative study, it must be taken in consideration the evaluation of each software, using some statistical measures like precision, recall, and F-Measure. Where Precision is the probability that a retrieved case is relevant, Recall is the probability that a relevant case is retrieved in a search, and F-Measure is the harmonic mean of recall and precision together which appeared in table 2.

Table 2 also shows a statistical view after applying the same query to the five CBR software. The results show that the highest accuracy reached and the number of cases retrieved and matched through the jCOLIBRI followed by myCBR, FreeCBR,CBR Shell and eXiTCBR respectively.

Table 2: CBR Shell Comparison

CBR Shell	Cases structure	selection strategies	Case retrieval	Case revised	Case storage	Case indexed	Graphical User Interface (GUI)	Dealing with uncertain data	Correct Matched Cases retrieved	Precision	Recall	F-Measure	Accuracy
CBR Shell	Textual	distance method	Two methods KNN Threshol	Manual	Text	No	Very simple GUI	Can’t handle	70	0.666	0.7	0.682	0.703

			d										
Free CBR	Textual	weighted Euclid distance	Simple matching	Manual	Text	No	Simple and easy but limited	Can't handle	81	0.764	0.81	0.786	0.813
jCOLIBRI	Xml /text	similarity functions	5 methods k-NN Threshold Ontology Textual OpenNLP and GATE Recommenders	Automatic	CSV XML	Yes	Simple and powerful Use wizard to simplify	Handle as null	99	0.933	0.99	0.961	0.994
myCBR	Object	similarity functions	Query model	Manual	CSV XML	No	user can customize the GUI and handle most of things	Handle as _unknown_ or _undefined_	93	0.902	0.93	0.916	0.934
eXiTCBR	Custom CSV	distance method or similarity measure	Simple Querying	Manual	Text	No	Very simple , no options	Can't handle	61	0.603	0.61	0.606	0.613

V. CONCLUSION

This paper introduces a comparison among most common used CBR software. It also mentions the advantages and disadvantages of each software. Moreover, this paper applies the same case base to the five CBR software to compare and evaluate the results using the predetermined factors and calculating Precision, Recall ,F-Measure and Accuracy for each one. As a conclusion CBR, Free CBR and eXit CBR are very simple software including simple GUI and only include the selection and retrieval of similar cases using traditional techniques. On the other hand, both myCBR and jCOLIBRI are more complex and can be used for complex CBR. myCBR interfaces over matches jCOLIBRI's and provides more options as weights and Similarity functions, type modification of attributes and cases. This is of great importance for query adjustment and refining the case base.

VI. FUTURE WORK

There should be a new technique to improve the CBR process, which needs to be developed and tested. The new idea is to use a semantic approach to store and retrieve cases with added meta-data to the cases itself to help in measuring similarity instead of the traditional techniques.

REFERENCES

[1] A. Aamodt and E. Plaza, “Case-based reasoning: foundational issues, methodological variations, and system approach” AI Communications 7(1), 39–59, 1994.

[2] Janet L. Kolodner, “An Introduction to Case-Based Reasoning” Artificial Intelligence Review 6, 3-34, 1992.

[3] Riesbeck, C. and Schank, R. (1989). Inside Case-Based Reasoning, Lawrence Erbaum Associates, Inc.

[4] Doyle M., Hayes, C., Cunningham, P. and Smith, B. (1998). CBR Net : Smart Technology over a Network. Department of Computer Science, Trinity College Dublin.

[5] Zhi-We Ni, Shan-Lin “Integrated Case-based Reasoning” Proceedings of the Second International Conference on Machine Learning and Cybernetics, XI; 2-5 November 2003.

[6] Leake, David, "CBR in Context: The Present and Future (http:/ / www. cs.indiana.edu/~leake/papers/p-96-01_dir.html/paper. html)", In Leake, D., editor, Case-Based Reasoning: Experiences, Lessons, and Future Directions. AAAI Press/MIT Press, 1-30, 1996.

[7] Simon C.K. Shiu, “Case-Based Reasoning: Concepts, Features and Soft Computing” Applied Intelligence 21, 233–238, 2004.

[8] http://www.aiai.ed.ac.uk/project/cbr/CBRDistrib

[9] http://freecbr.sourceforge.net/

[10] Bello-Tomás,J.J., González-Calero, P. A., Díaz-Agudo, B. (2004): JColibri: An Object-Oriented Framework for Building CBR Systems. In Advances in Case-Based Reasoning, Lecture Notes in Computer Science. Springer Berlin/ Heidelberg, Vol. 3155/2004, p. 32-46.

[11]K.Boshnakov,C.Boishina,M. Hadjiiski, Multiagent fault-tolerant supervising control of wastewater treatment plants for wastewater, International Conference,Automaticsandinformatics'11,Bulgaria, Sofia,2011.

[12] M.Hadjiski, V.Boishina,EnhancingFunctionality of Complex Plant Hybrid Control System UsingCase-Based Reasoning,IntelligentSystems(IS), 20105th IEEE International Conference, 7-9 July 2010, London,25-30.

[13] http://www.mycbr-project.net/index.html

[14] http://exitcbr.udg.edu/

[15]http://exit.udg.edu/download.aspx?id=13eXiT*CBR user tutorial - eXiT*CBR 3.0 user tutorial

[16] http://archive.ics.uci.edu/ml/