

# A CBR Decision Support System for Loan Evaluation

Noureddine Benmessaoud<sup>1</sup>, Abdelkader Adla<sup>2</sup>

<sup>1</sup>Department of Computer Science  
University of Oran

<sup>2</sup>Ahmed Ben Bella  
Oran, Algeria

[adla.abdelkader@univ-oran1.dz](mailto:adla.abdelkader@univ-oran1.dz)



*Journal of Digital  
Information Management*

**ABSTRACT:** *Group decision making is time-consuming, complex, critical and recurring in nature. In group decision making, the decision-makers often perform periodic and repetitive activities. This prompts the decision makers to store their knowledge and their experience invoked in prior problem solving sessions. The development of this group knowledge clearly assists in similar later decision-making activities and reduces the time required to come to a decision, particularly, in a critical situation. This shared knowledge constitutes the group memory. In this paper, we develop a Case-Based Reasoning Decision Support System called CBR-DSS that makes use of the rules, policies and standard procedures of prior groups embedded in the group memory. The CBR process is evolved by using the Analytic Hierarchy Process (AHP). In an effort to enhance utility and show the feasibility, we experiment our approach in a bank loan analysis domain, a semi-structured multiple attributes decision-making environment, and compare CBR-DSS with a conventional case based reasoning system. CBR-DSS will compensate for lack of experience of young bank consultants, and allow disseminating and distributing available experience to different bank sites.*

## Subject Categories and Descriptors

[H.3] Information Storage and Retrieval; [I.2] Artificial Intelligence

**General Terms:** Group memory, Information Search, Decision support systems, Case-Based Reasoning

**Keywords:** Recommender System, Case-based Reasoning, Information Retrieval, Group Memory, Group Decision Making

**Received:** 22 October 2018, Revised 19 December 2018, Accepted 4 January 2019

**Review Metrics:** Review Score 4.95/6, Revise Scale: 0-6, Inter-reviewer Consistency: 77.5%

**DOI:** 10.6025/jdim/2019/17/2/75-86

## 1. Introduction

The decision making processes involve decision makers performing periodic and repetitive activities. This prompts them to store their knowledge and their experience invoked in prior problem solving sessions in a shared repository. This shared repository together with the appropriate means for managing its content is in fact a utilization of group memory. The group memory, that stores the knowledge of group members and experience, retains the rules, policies and procedures of an organization, is useful for decision makers engaged in similar decision-making activities, and will clearly assist them in similar later decision-making activities and reduces the time required to come to a decision, particularly, in a critical situation.

Previous research has proposed the use of group memory to accumulate, organize, preserve, link and share diverse knowledge coming from past experiences [1][2][3][4][5]. The development of a Group memory is seen as the means by which knowledge from the past can be used to influence present organizational activities.

The group memory and its associated mechanisms to capture the experiential knowledge of these groups can

be of significant value to the organization in general, and the group in particular.

The advantages of capturing group memory are many, including simplification of the process, codification of decision strategies, and carryover of knowledge when group composition changes, increased organizational learning, [6] [7] [8], lowered transaction costs to implement the chosen decision, reduced time to making decision choices, etc. in addition, the group memory can be used to increase group effectiveness by supporting the coordination of work group, management of information, organization's responsiveness to changes, and pursuit of organizational goals [9]. Other cases where such a memory will be of utility are multi-stage decision making and tasks that cannot be satisfactorily completed in a single session, creating a need to carryover intermediate findings to subsequent sessions.

While there has been substantial research related to group decision-making and group memory concept mentioned in literature [10][11], the explicit use of group memory to support repetitive decision-making has received less attention and no major implementation of memory has actually occurred. Mostly decision models are only indirectly supported and information is not captured and stored in a structured representation limiting retrieval and reuse of decisions that is easily shared.

As a way to deal with these needs, we propose a recommender system called CBR-DSS for a bank to access and retrieve group knowledge in a similar domain. The integration of a group memory in the form of a case base within a group decision support system is likely to provide additional information processing support. The Group memory indexes and retrieves cases to propose whole or part of them as solution(s) to a new problem [12]. In this way, it increases knowledge reuse functionalities through cases and their components, and allows reducing the time required to come to a decision, particularly, in a critical, compensating for lack of experience of young bank consultants, and disseminating and distributing available experience to different bank sites.

The remaining part of the paper is organized as follows. First, we outline in section 2 the role of the group memory in organizations, its use in group decision making, and the case-based reasoning paradigm. Next, we describe our group memory approach to case-based reasoning decision support system and its integration within group decision support systems. Finally, we describe a bank loan evaluation, a case study to illustrate the feasibility of our idea before concluding.

## 2. Background

### 2.1 Role of Group Memory in Group Decision Making

Group Decision Making (GDM) focuses on group activities or tasks which involve a sequence of operations and actors [13]. GDM researchers have employed shared repositories

such as knowledge bases and e-mail folders to store group level information and knowledge from group solving sessions. GDM literature has frequently used the term "group memory" for this shared repository and its tools.

Group decision making can greatly benefit from group memory. Such a memory can support a group's work within and across sessions. It can also provide uniform, consistent knowledge acquired from prior sessions and can be used "to bring new participants up to speed by browsing through knowledge about current session and former sessions" [10]. Group decision systems based on group memory support the capture, storage, and recall of group memory [14].

Group memory has a significant role in intelligent decision support. It is assumed that intelligent decision support should provide some memory aids for the decision maker and include "learning" from the decision maker's experience. Group memory allows the team to resume deliberations at relatively little cost and no significant rework of prior material. Change in group membership also needs access to group memory. New members must be initiated in the group process and existing members may take on additional or different roles. An iterative process, where group members work through multiple rounds to achieve consensus, could also benefit from group memory.

Group memory integrated with a group decision support system should be designed as means for providing easy access and retrieval of relevant information to decision makers. Case-Based Reasoning (CBR) systems are the most prevalent ones that implement group memory. The case-based reasoning formalism was proposed as a way of reflecting human knowledge by storing data about significant experiences as "cases" and manipulating reasoning by analogy.

### 2.2 Case-Based Reasoning

Case-Based Reasoning (CBR) originated at the end of 70s when first Schank and Abelson laid the foundations for the creation of CBR [15]. The case-based reasoning formalism was proposed as a way of reflecting human knowledge by storing data about significant as experiences "cases" and manipulating reasoning by analogy.

In CBR terminology, the central notion is a case which is a contextualized piece of knowledge representing a previous experience. A *case* usually denotes a *problem situation* which has been captured and learned in a such way that it can be reused to solve future problems. It is referred to as a past case, previous case, stored case, or retained case. Correspondingly, a new case or unsolved case is the description of a new problem to be solved.

Over the last few years CBR as an alternative reasoning paradigm and computational problem solving method has increasingly attracted more and more attention. It uses the fact that humans often solve new problems by comparing similar problems that they already know how

to solve. The main principle of CBR is: *similar problems have similar solutions*. Reasoning by re-using past cases is a powerful and frequently applied way to solve problems for humans.

In many respects CBR is fundamentally different from other major AI approaches. A CBR system draws its knowledge from a reasonably large set of cases contained in the case library of past problems rather than only from a set of rules. Instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, the CBR approach collects information about previous cases, and then retrieves this information for similar cases. By adopting this approach, it is able to utilize the specific knowledge of previously experienced, actual situations. Subsequently, the previous solutions may be adapted so that they more closely match the current problem and situation.

There have been various models to represent the CBR method [16]-[17]-[18]. Currently, there is a general acceptance of R4 model introduced by Aamodt and Plaza [19] extended by others with the well-known cyclical process comprising the five stages (Figure. 1):

1. *REPRESENT* and indexing cases in the case base;
2. *RETRIEVE* a subset of cases from the library that are relevant to the client's needs and circumstances and the most similar case(s);
3. *REUSE* the solutions with best potential to meet client's needs and to attempt to solve the problem;
4. *REVISE* the proposed solutions with modifications if necessary to develop a negotiated treatment plan meaningful to the client; and

5. *RETAIN* the new solution as a part of a new case, if it is useful for future problem-solving. This R5 model is more complex and deeper than this mere presentation because each step involves a number of more specific sub-processes with their own difficulties.

Systems relying on that paradigm have been successfully used in several application domains, such as diagnosis, classification, prediction, control and action planning. Various reasons have contributed to the attractiveness of employing case-based methods: They are straightforward to implement, help in reducing the knowledge acquisition effort and they are noise-tolerant due to their approximate nature. However, there is a few on integrating CBR into the architecture of group decision making systems. Nunamaker et al. [20] mention group memory as a component in the University of Arizona's GroupSystems, an electronic meeting system. Weiser and Morrison [8] describe the use of a project memory that captures, retains, and indexes project information so that people external to the project can use it later.

### 3. A group Memory Based Approach to Support Group Decision Making

Using group memory, the problem decision process should involve the representation and utilization of knowledge represented by cases, and adopts a group memory based reasoning approach. This approach emphasizes a collection of cases as a group memory, and reasoning as a process of accessing and searching in this memory.

#### 3.1. Case Representation and Indexing

The main goal of the case representation is to organize the knowledge needed in a relevant way to identify the main characteristics describing a problem and its associated solution and to ensure the retrieval of the most

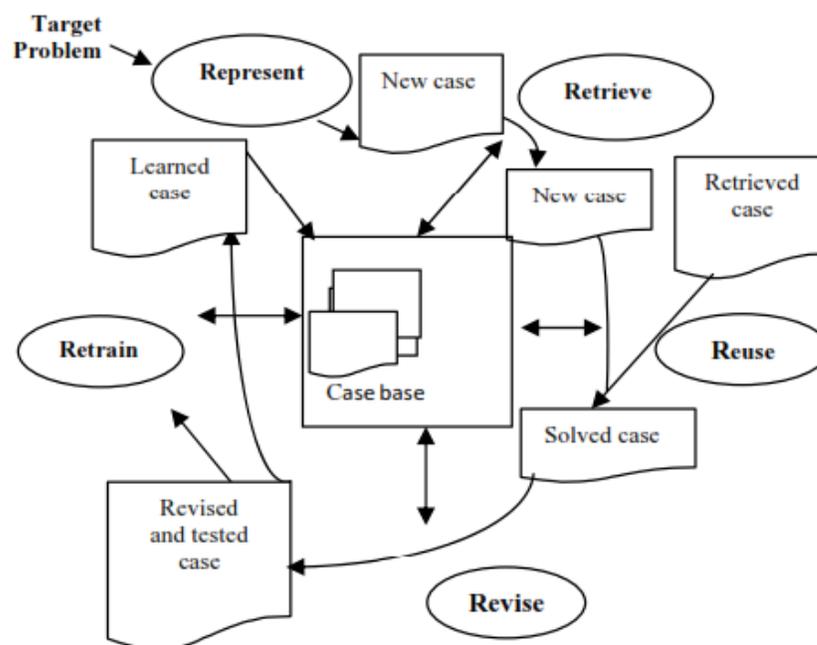


Figure 1. R5 CBR Cycle

appropriated cases. A case includes decision-related information such as tasks, method(s), decision issue, criteria and arguments [21].

**Name:** Task name

**Parameters:** Parameter list, it represents the set of world objects (described in the domain model) handled by the task

**Objective:** Task's Goal described as a state of the world after task performance

**Methods:** List of Methods achieving the task

To each task are associated the methods which are a priori declared as the best adapted to achieve it. A method represents a resolution mechanism or the know-how which it is possible to implement in order to achieve a task. A method is characterized by the following fields:

**Name:** Method name

**Task:** Task achieved by the method

**Applicability conditions:** Applicability conditions needed for the instantiation of the method

**Preconditions:** Preconditions that must be satisfied to apply the method

**Effects:** Effects generated by the successful application of the method

**Control:** Achievement order of the sub-tasks

**Sub-tasks:** Sub-tasks set

Several approaches can be applied to organize cases for efficient retrieval. A flat case base is the most simple and common organization. Here, all the cases are stored at the same level. The major drawback of the flat case base remains the exhaustive search through the whole memory. Moreover, this drawback increases sharply when the number of both features and cases increase which requires a huge computational effort. The case-base is organized into a manageable structure that supports efficient search and retrieval methods. Using a decision tree structure, a balance is found between storing methods that preserve the semantic richness of cases and their indices and methods that simplify the access and retrieval of relevant cases.

Case indexing and storage are an important aspect in designing efficient CBR systems in that, it should reflect the conceptual view of what is represented in the case and take into account the indices or features that characterize the case and help in distinguishing one case from another. Case indexing refers to partitioning the group memory for a faster retrieval of relevant subset of cases. Case indexing involves assigning indices to cases to facilitate and ensure that they can be retrieved at appropriate times. Each case in the group memory is

assigned a specific label which identifies under what conditions the case may be useful, and also suggests the applicability of the case to a new one when retrieved. The case descriptors or attributes can be either of nominal type or numeric one.

Instead of an exhaustive search, we use the indexing trees. Trees result in recursively partitioning a dataset into subsets. In such a hierarchical structuring approach, index cases are structured into categories to reduce the number of available cases for similarity measurement. Indexing trees also enable richer case representations of past situations than a simple database, since a case includes information about the context of the event as well as details.

The number of cases in the case base is going to grow due continuous memorization of new transactions. In order to reduce the research time and to increase the effectiveness of the retrieval process, the latter is decomposed in two steps: The first one consists in selecting a subset of relevant source cases whilst the second one is dedicated to the similarity measurement and the ranking of source cases included in the subset.

### 3.2. Case Retrieval

The operation of the similarity measure is to establish similarities between the cases. More precisely, it is to calculate the degree of similarity between the source cases and the target case. This operation is carried out over two stages: 1) Measuring local similarities between the characteristics of a source case and those of the target case by using a similarity operator; 2) Measuring global similarity from the local similarities; the global similarity is as the quotient of the sum of the local similarities by the sum of the corresponding weights of the characteristics.

The local similarities are used to compute similarities between values of single attributes. They are calculated for each attribute  $i$  by comparing the value of the target case  $x_i$  with the corresponding source one  $y_i$ . However, as problem features are described by different types of values (nominal or numeric), the local similarity is calculated regarding these types.

#### 3.2.1. Local Similarity Measurement for Numerical Features

For numerical values, the local similarity is often calculated with a distance measurement. Various distances have been proposed to measure the variation between two values. The most used ones are the Euclidian and the Manhattan distances which are in fact particular kinds of the Minkowski measurement. Eq. (1): For two cases  $X$  and  $Y$ :

$$d(X, Y) = (\sum_{i=1}^L w_i |x_i - y_i|^p)^{1/p} \quad (1)$$

Where  $x_i$  and  $y_i$ , represent respectively the  $i$ th feature of  $X$  and  $Y$ , and  $w_i$  the associated weight to this feature.

As the global distance cannot be calculated due to the different features values types, we get a local distance for each feature. For numerical features, the calculation of the local distance is based on the following equation Eq. (2): (derived from Eq. (1)):

$$d(x_p, y_i) = |x_i - y_i| \quad (2)$$

However, this way of measuring can distort the results when the features have different definition domain sizes. Thus, to normalize the distance calculation, we introduce the  $Int_i$  function to explicitly express the definition domain. This function expresses the difference between the maximum and the minimum values for the feature  $i$ .

$$d(x_p, y_i) = \frac{|x_i - y_i|}{int_i} \quad (3)$$

Finally, the local similarity for a numerical characteristic can be calculated from the distance given by equation (3).

According to equation (4), two closest problems are the most similar.

$$sim(x_p, y_i) = \frac{|x_i - y_i|}{int_i} \quad (4)$$

### 3.2.2. Local Similarity Measurement for Nominal Features

For nominal features, the classical local measurement considers two values Eq. (5):

$$sim(x_p, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{if } x_i \neq y_i \end{cases} \quad (5)$$

So, if two features  $(x_p, y_i)$  have the same nominal value, then they are identical (similarity = 1), otherwise they are different (similarity = 0).

### 3.2.3. Global Similarity

The global similarity is calculated from all the local similarities in order to establish similarities and more precisely the degree of similarity between the target problem and source ones. The target problem(X) is compared with a source problem(Y) in the case base by the global similarity measurement Eq. (6):

$$sim(X, Y) = \frac{\sum_i^n w_i sim(x_p, y_i)}{\sum_i^n w_i} \quad (6)$$

With the weight  $w_p$ , the user can customize the global similarity weighting one feature more importantly than the others. This choice is crucial to obtain relevant similarity measurement.

The global similarity criteria allow ranking all the source cases from the most similar to the less one. The former are used to express the different importance between features. The user can assign the weight values, or rank

the attributes upon their importance. Attribute having the rank of 1 is the most important, and two attributes can have the same rank. For each attribute, the corresponding weight is calculated by Eq. (7):

$$w_i = 1 - \frac{R_i - 1}{Max(R_i)} \quad (7)$$

Where:

- $w_i$  is the attribute weight
- $R_i$  is the importance rank of the attribute in the CBR system
- $Max(R_i)$  is the maximum value of importance between the attributes

### 3.2.4. Assigning features weights using the Analytic Hierarchy Process (AHP)

Retrieving similar cases to a new case involves determining the importance of each attribute in relation to other attributes, expressed in terms of weight. To do this, we use the Analytic Hierarchy Process (AHP) method to classify attributes, calculate their weights and measure the similarity between cases.

The AHP is a popular Multi-Criteria Decision-making Method (MCDM) developed by Saaty in the 1970s [22], designed to solve complex multi-criteria decision problems. It is a flexible and powerful tool for handling both qualitative and quantitative multi-criteria problems. It has been widely used for ranking a finite set of alternatives and for choosing the best alternative from a finite set of alternatives [23]. The AHP is aimed at integrating different measures into a single overall score for ranking decision alternatives based on pairwise comparisons on a scale 1 to 9 in a matrix A [22].

To establish priorities, weights are assigned to each attribute. These weights are assigned through a process of comparative judgements called pairwise comparison. In pairwise comparison, each attribute is compared at a peer level in terms of importance. Priorities are calculated from the comparison matrix A by the eigenvalue method:

$$A * w = \lambda * w \quad (9)$$

Where  $A$  is the comparison matrix,  $w$  is the vector of the priorities,  $\lambda$  is the maximal eigenvalue.

Consistency analysis must be performed [22] to ensure the judgmental consistency of the matrices, that is to say compliance with the transitivity rule in terms of a consistency ratio lower than a typical threshold value of 0.1.

It should be noted that the weight in the AHP is different from the one we use for the similarity measure. The latter, as we have already mentioned, is calculated by equation (8) where  $R_i$  is the rank of importance of each attribute  $i$ . In this sense, the weight vector  $w$  given by the AHP method

is exploited as a vector of importance rank relative to the attributes.

#### 4. Application to Bank Loan Assessment

We experiment our approach on a case of bank loan analysis for a bank loan for properties.

Properties are heterogeneous in the sense that price is a composition of effects of various characteristics. The diversity of immovable characteristics requires the search for a similar dataset. This technique is known as the method of comparing prices on the market for similar properties. The quality of properties valuation depends on the appropriate selection of the sample.

The availability of data has enabled the creation of databases with thousands of cases, for example in public administrations (sales taxes) and banks (real estate financing). In this situation, there is data availability, but the selection of relevant cases is not a trivial task. Manual selection is impractical, due to the possibility of error in the selection and the time spent by the evaluator and the resulting cost. In addition, the availability of data allows the use of artificial intelligence tools. When choosing techniques, it is important to take into account property valuation requirements, such as the accuracy of estimates and the ease of explanation of the models used.

The property appraisal process has some subjectivity, with the evaluator's influence on the case selection process and the attributes of the appraisals. In this case, developing a more objective, systematic data analysis format can help increase accuracy.

Given the importance of valuations and the property market, it is important to look for alternatives to improve valuations. This article proposes the use of case-based reasoning for case selection and evaluation.

To consent for a bank loan for a customer for housing, the bank must take into account some lineaments allowing its real price assessment so that if the customer would be unable to repay the loan, the bank can sell the land or housing with its price. The latter varies from one area of land to another. It depends on number of lineaments such as façades of the plot of land and whether is built or not. This makes the banking operation complex and various phenomena may occur. Therefore, the bank must consider all these parameters to compare the amount of the loan requested by the customer and the real price of the land or housing and give the right decision: accepting or refusing the loan demand.

The bank receives lot of loan requests from customers. To assist bank financial consultant in loan analysis and endow her/him with tools aiding her/him to make quickly right decisions without comparing and analyzing each time all the characteristics of the land or housing, we propose to develop and use the group memory based decision

making system, called CBR-DSS.

CBR-DSS supports the bank financial consultant to make his analysis based on knowledge and past experiences of experts. The bank organizational memory is a case base where each case consists of an already analyzed loan demand with its features and its associated decision.

In order to implement our system, we have collected the data from the "Cadastre" office (land register). The latter manages the land and buildings of the sites built or not, urban or rural through the national territory. The collected data serve to analyze and manage the bank loans demands in order to finance the granting of a ground or housing by customers. Based on these cadastral data, we performed a simulation for a bank wanting to manage bank loans. We considered a sample of 122 land properties from the Cadaster Directorate of Oran department, all types combined (rural or urban, built or not built, one or two façades, or even three, etc.).

To evaluate our system and assess the system performance, we have carried out a number of experiments. Our system will be also compared with a conventional CBR called *FreeCBR* (a free open source Java implementation of a Case Based Reasoning tool) authored by Lars Johanson [24]. Table I presents an example of a loan request.

Loan Requested (Millions of DA)	Property				
	Type of the Plot	Area (m <sup>2</sup> )	Zone of de façades	Number (%)	Conditions
5	Built	350.50	Urban	1	70

Table 1. Example of a bank loan request

#### 5. Comparative Study

The Figures 2 and 3 show the results of relevant cases retrieving with both systems. It should be noted that *FreeCBR* retrieves all cases, from the case base, with different similarities. On the other hand, our system only retrieve cases with a similarity greater than or equal to 0.5 ( $\geq 0.5$ ) to the new case. We harmonized the case retrieving process for both systems by considering only cases with a similarity of 0.5 ( $\geq 0.5$ ).

Specifically, as shown in Tables 2.a and 2.b, we calculated some values such as A the number of relevant cases retrieved, B the number of non-relevant cases retrieved, C the number of relevant cases non-retrieved by the system, and D the number of non-relevant cases nonretrieved:

To evaluate the effectiveness of the search strategies, we use statistical measures such *Precision*, *Recall*, *Accuracy* and *F-Measure*.

Hit %	land	area	zone	sides	cost	state	decision	N° Case
96.42576894280096	Built	467.7	Urban	1	550.7	35	Unfavorable	37
95.66386459514225	Built	212.15	Urban	1	567.89	25	Favorable	76
95.63322310613666	Built	284.74	Urban	1	370.57	35	Favorable	44
95.39515610907286	Built	355.44	Urban	1	650.0	80	Favorable	26
94.89966940482053	Built	202.48	Urban	1	600.5	75	Favorable	45
94.19337325678185	Built	196.22	Urban	1	370.5	50	Favorable	102
93.71413775788048	Built	131.8	Urban	1	560.78	56	Favorable	95
90.75251147515787	Built	383.25	Urban	1	800.0	47	Unfavorable	46
85.57906338642701	Built	358.84	Urban	1	970.0	75	Favorable	27
85.54075988019514	Built	629.1	Urban	1	900.0	25	Unfavorable	47
83.39331178492206	Built	629.1	Urban	1	980.5	75	Unfavorable	72
55.261750443304415	Built	353.09	Rural	1	460.0	35	Unfavorable	23
55.24061024630056	Not built	354.91	Urban	1	560.0	50	Unfavorable	29
55.22975432511039	Built	345.0	Rural	1	568.0	50	Unfavorable	70
55.22489817135238	Built	392.36	Urban	2	560.89	45	Unfavorable	79
55.20721981707263	Not built	260.89	Urban	1	480.9	80	Favorable	117
55.19320070029721	Not built	279.36	Urban	1	560.0	55	Unfavorable	42
55.173316918758466	Built	355.59	Urban	2	600.0	40	Unfavorable	28
55.16994342831668	Built	267.0	Rural	1	569.0	35	Favorable	66
55.161763357695	Built	239.31	Rural	1	465.0	55	Favorable	10
55.16127972962519	Built	259.62	Urban	2	567.5	60	Unfavorable	3
55.14033846706631	Built	227.57	Rural	1	467.34	90	Favorable	86
55.11931771768255	Built	235.0	Rural	1	567.0	55	Favorable	67
55.101775961484066	Built	345.67	Urban	3	370.3	30	Unfavorable	106
55.0456683998661	Built	202.27	Rural	1	567.89	35	Unfavorable	88
55.03833947555257	Not built	185.22	Urban	1	467.9	34	Unfavorable	75
55.003265445248924	Not built	526.77	Urban	1	470.0	20	Unfavorable	40
54.97293944394476	Not built	276.7	Urban	1	657.45	45	Favorable	73
54.94556810691154	Built	360.86	Urban	2	678.0	56	Unfavorable	101
54.94121137253627	Not built	412.66	Urban	1	670.57	36	Favorable	96

Figure 2. Cases retrieved with FreeCBR

NUM	LAND	AREA	ZONE	SIDES	COST	STATE	DECISION	SIMILARITY
116	Built	145,45	Urban	2	500	56	Favorable	0,664
30	Built	348,23	Urban	2	780	85	Unfavorable	0,65
106	Built	345,67	Urban	3	370,3	30	Unfavorable	0,626
26	Built	355,44	Urban	1	650	80	Favorable	0,625
28	Built	355,59	Urban	2	600	40	Unfavorable	0,624

Number of retrieved cases **28**

Figure 3. Cases retrieved with CBR-DSS

	CBR-DSS	FreeCBR
<b>A</b>	9	14
<b>B</b>	19	51
<b>C</b>	8	3
<b>D</b>	86	54
<b>A+B</b>	28	65
<b>C+D</b>	94	57
<b>A+C</b>	17	17
<b>A+D</b>	95	68
<b>B+D</b>	105	105
<b>A+B+C+D</b>	122	122

Table 2(a). Comparing Case retrieved

		Relevant	Irrelevant	Total
<b>Retrieved</b>	<b>CBR-DSS</b>	A : 9	B : 19	A+B : 28
	<b>FreeCBR</b>	A : 14	B : 51	A+B : 65
<b>Notretrieved</b>	<b>CBR-DSS</b>	C : 8	D : 86	C+D : 94
	<b>FreeCBR</b>	C : 3	D : 54	C+D : 57
<b>Total</b>	<b>CBR-DSS</b>	A+C : 17	B+D : 105	A+B+C+D : 122
	<b>FreeCBR</b>	A+C : 17	B+D : 105	A+B+C+D : 122

Table 2(b). Comparing Case retrieved

There is an inverse relationship between precision and recall. Increasing precision often involves reducing recall. The inverse relationship between precision and recall compels the system to come to a compromise between them. To this end, F-Measure (F-score) is calculated by Eq. 12:

$$Precision = \frac{A}{A+B}$$

$$Recall = \frac{A}{A+C}$$

$$Accuracy = \frac{A+D}{A+B+C+D}$$

$$F - Measure = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$$

All these measures are calculated for both systems (Table 3) in order to evaluate their performance.

We ran five queries on both systems. The results of retrieving relevant and irrelevant cases with both systems are summarized in (Table 4). The analysis of these results leads to a comparisons between the two systems.

	Precision	Recall	Accuracy	F-mesure
<b>CBR-DSS</b>	0,321	0,529	0,779	0,4
<b>FreeCBR</b>	0,215	0,824	0,557	0,341

Table 3. GMSS and FreeCBR comparison based on statistical values

Loan Requests	CBR-DSS			FreeCBR		
	A+B	A	D	A+B	A	D
Req. 1	28	9	86	65	14	51
Req. 2	7	3	101	63	17	59
Req. 3	7	3	108	55	10	67
Req. 4	24	7	93	59	10	61
Req. 5	7	5	109	19	10	102

Table 4. Performance Thresholds

		<b>CBR-DSS</b>	<b>FreeCBR</b>
Precision	Req. 1	0.321	0.215
	Req. 2	0.429	0.270
	Req. 3	0.429	0.182
	Req. 4	0.292	0.169
	Req. 5	0.714	0.526
Recall	Req. 1	0.529	0.824
	Req. 2	0.176	1
	Req. 3	0.300	1
	Req. 4	0.583	0.833
	Req. 5	0.455	0.909
Accuracy	Req. 1	0.779	0.557
	Req. 2	0.852	0.623
	Req. 3	0.910	0.631
	Req. 4	0.820	0.582
	Req. 5	0.934	0.918
F-measure	Req. 1	0.400	0.341
	Req. 2	0.250	0.425
	Req. 3	0.353	0.308
	Req. 4	0.389	0.282
	Req. 5	0.556	0.667

Table 5. Performances Comparison based on statistical measures

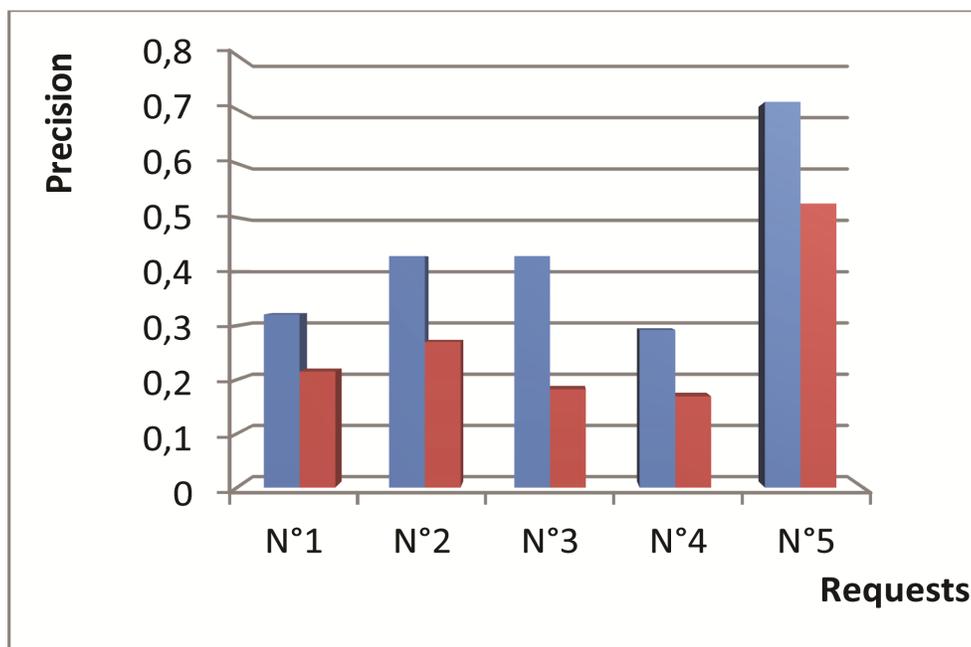


Figure 4. Precision

The FreeCBR system retrieves more relevant cases (the A number) than CBR-DSSSSS. But on the other hand, CBRDSS retrieves much less irrelevant cases (the number D) than FreeCBR. That said, CBR-DSS retrieves a few less relevant cases compared to FreeCBR, but it eliminates many more unnecessary irrelevant cases.

The two systems are also compared against the four calculated statistical measures (Table 5).

• **Precision and recall:** To see the difference between the two systems, the values of these two parameters are shown graphically. Then, as depicted in (Figure 4), we can see that CBR-DSS has more precision than FreeCBR.

In other words, GMS is more accurate in recovering relevant cases than FreeCBR. On the other hand, CBR-DSS has missed more relevant cases in the collection of relevant cases, in general, which is measured by recall, figure (Figure 5).

**F-measure:** The inverse relationship between precision and recall, mentioned by F-measure forces to a compromise: some tasks particularly require good precision whereas others need good recall. Out of the five tests, CBR-DSS has, in three times, an F-measure higher than that of FreeCBR, i.e. CBR-DSS makes a good compromise between precision and recall than FreeCBR. (Figure 6)

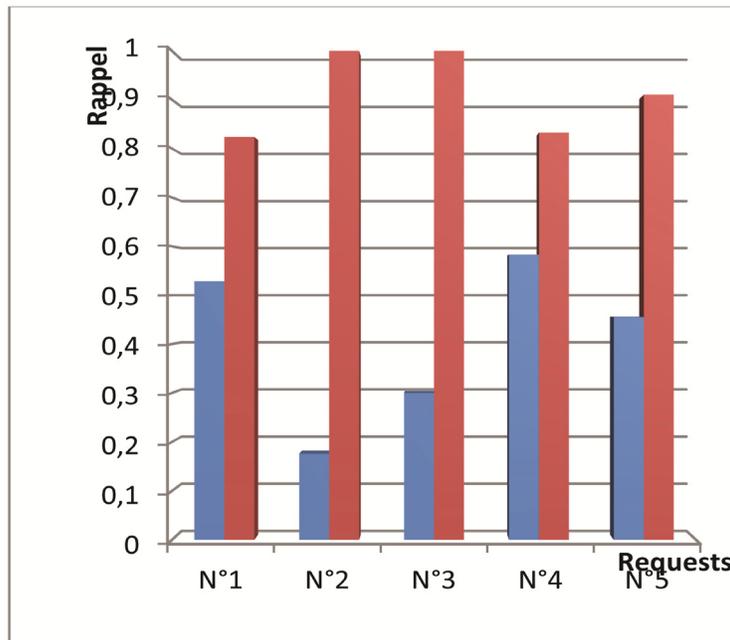


Figure 5. Recall

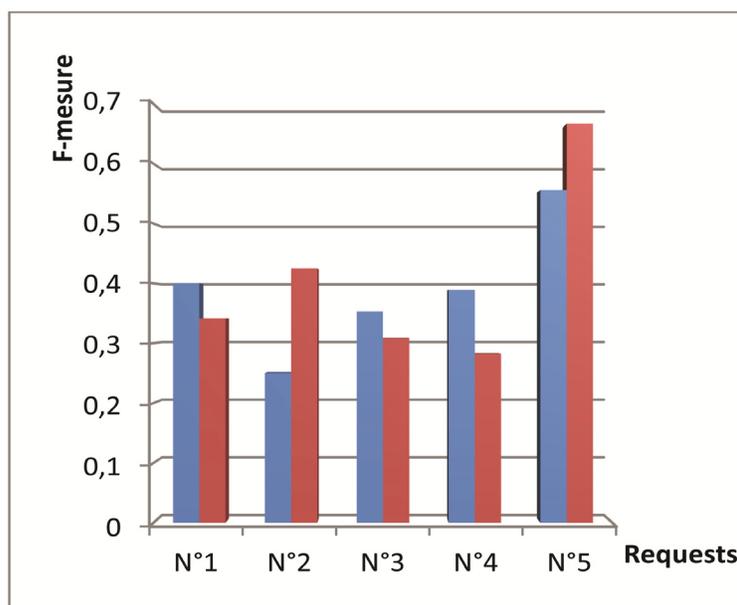


Figure 6. F-Measure

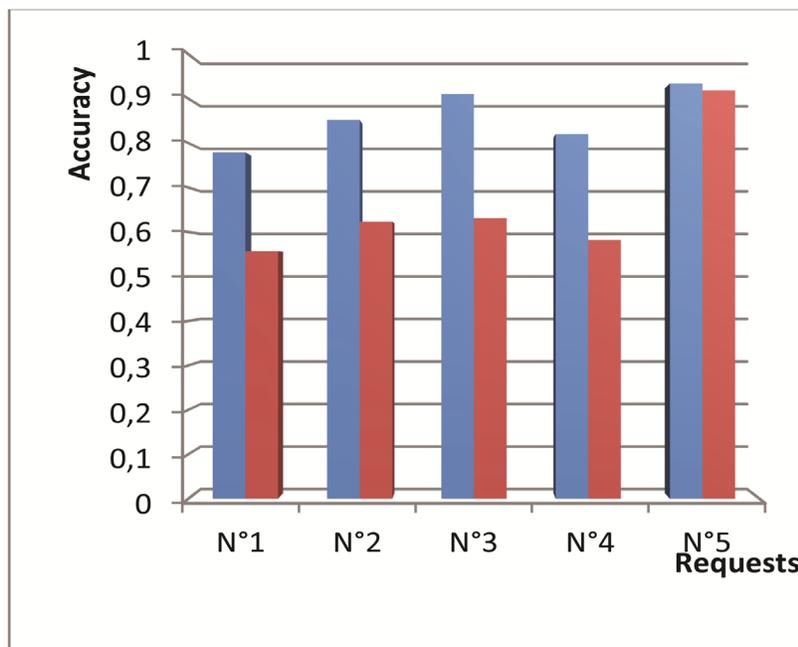


Figure 7. Accuracy

• **Accuracy:** As depicted in (Figure 7), out of the five tests, CBR-DSS has a higher accuracy than FreeCBR. This finding means that CBR-DSS has the highest proportion of correctly classified cases either relevant or not.

In current bank activity, we must come to an appropriate decision in a short time. It's a precision-critical task where retrieving just one relevant and more similar case to the target problem is sufficient. That means a full recall of all relevant cases is not required. In other words, we need a quick search of the relevant cases to make promptly the decision. So, we need a precision in the relevant cases and a higher recall, i.e. a higher F-measure. Regarding all the previous results, the proposed CBR system outperforms FreeCBR, the conventional CBR. Nevertheless, more experiments and comparisons with other systems are needed to establish the performance of our system.

## 6. Conclusion

Group Memory Systems have been recognized as means for supporting the collection, storage and distribution of the knowledge and experience of the organization. These systems are proposed to improve the quality of group decision-making processes by using the power of recent computer technologies.

We described in this paper a group memory-based reasoning approach. The use of this reasoning mode enables decision makers to reduce the time required to come to a decision, particularly, in a critical situation. We applied our approach on bank loan analysis to provide a bank consultant with access to past experience in the form of cases using a reasoning mechanism to retrieve the relevant cases.

It appears evidence that this form of system is useful for intelligent decision support. Overall the experiment showed in this study the Group Memory System using CBR to search and recall previous decisions which are in some way "similar" to the current decision situation improve the efficiency of performance of the bank consultant.

Finally, in order to evaluate the proposed system, a comparative performance was undertaken with FreeCBR, a conventional CBR, using statistical parameters. We found that the CBR-DSS system shows better values in relation to some parameters and outperforms the conventional FreeCBR particularly in in relation to precision and accuracy measures. Besides, a new index method is used in order to facilitate the retrieval of relevant cases using the decision tree index and reducing thus the search time.

In future work, we plan to improve CBR-DSS in the case retrieving step to allow calculating and retrieving the similar cases and choosing the most appropriate and relevant source case(s) by coupling the traditional similarity measure with the fuzzy set theory. Also we intend to integrate CBRDSS in the GDSS framework [25][26][27].

## References

- [1] Gallupe, B. (2001). Knowledge management systems: surveying the landscape, *International Journal of Management Reviews*, 3 (1) 61-77.
- [2] Minhyung, K., Byoungsoo, K. (2017). Motivation, opportunity, and ability in knowledge transfer: a social network approach, *Knowledge Management Research & Practice*, 15 (2) 214-224.
- [3] Connelly, C. E., Ford, D. P., Turel, O., Gallupe, B.,

- Zweig, D. (2014). I'm busy (and competitive)!, Antecedents of knowledge sharing under pressure, *Knowledge Management Research & Practice*, 12 (1) 74-85.
- [4] Ramos, I., Lavina, L. (2014). Organizational Memory: A Preliminary Model Based on Insights from Neuroscience, In C. Machado and J.P. Davim (Eds), *Transfer and Management of Knowledge*, p. 167-205, 2014.
- [5] Akoumianakis, D. (2009). Practice oriented toolkits for virtual communities of practice, *Journal of Enterprise Information Management*, 22 (3) 317, 2009.
- [6] Ackerman, M. S., Halverson, C. A. (2000). Reexamining organizational memory, *Communications of the ACM*, 43, (1) 58-64.
- [7] Jackson, P. (2012). Transactive directories of organizational memory: Towards a working data model, *Journal: Information & Management*, 49 (2) 118.
- [8] Weiser, M., Morrison, J. (1998). Project memory: Information management for project teams, *Journal of Management Information Systems*, 14 (4) 149-166, 1998.
- [9] Stein, E.W., Zwass, V. (1995). Actualizing organizational memory with information systems, *Information Systems Research*, 6 (2) 85-117.
- [10] Hasemana, W. D., Nazaretha, L. D., Paulb, S. (2005). Implementation of a group decision support system utilizing collective memory, *Information & Management*, 42, p. 591-605, 2005.
- [11] Burstein, F., Jennex, M. E., Olfman, L. (2003). Organizational Memory, *Handbook on Knowledge Management*, Edited by C. W. Holsapple, Springer, Berlin - Heidelberg, p. 207-234, .
- [12] Stein, E.W. (1995). Organizational memory: review of concepts and recommendations for management, *International Journal of Information Management*, 15 (1) p. 17– 32, 1995.
- [13] Hibbard, J. (1997). Knowledge management—knowing what we know, *Information Week*, 1997.
- [14] Ackerman, M. S., Halverson, C. (2004). Organizational Memory as Objects, Processes, and Trajectories: An Examination of Organizational Memory in Use, *Computer Supported Cooperative Work (CSCW)*, 13 (2) 155.
- [15] Schank, R., Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding*, Hillsdale, N.J.: Lawrence Erlbaum, 1977.
- [16] Kolodner, J. (1993). *Case-Based Reasoning*, Morgan Kaufmann Publishers, Inc., 1993.
- [17] Aamodt, A., Plaza, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *AI Communications*. IOS Press, 7 (1) 39-59.
- [18] Charest, M., Delisle, S., Cervantes, O., Shen, Y. (2008). Bridging the gap between data mining and decision support: A case-based reasoning and ontology approach, *Intelligent Data Analysis*, 12 (2) 211-236.
- [19] Bergmann, R., Kolodner, J., Plaza, E. (2006). Representation in case-based reasoning, *Knowledge Engineering Review*, 20 (3) 209–213.
- [20] Nunamaker, J. F., Briggs, R., Mittleman, D., Vogel, D. R., Bakhtzard, D. (1997). Lessons from a dozen years of group support systems research, *Journal of Management Information Systems*, 13(3) 163.207.
- [21] Adla, A., Soubie, J. L., Zaraté, P. (2007). A cooperative Intelligent Decision Support System for Boilers Combustion Management based on a Distributed Architecture, *Journal of Decision Systems (JDS)*, 16 (2) p.241-263.
- [22] Saaty, T. (1977). A scaling method for priorities in hierarchical structures, *Journal of Mathematical Psychology*, 15 (3) 234-281.
- [23] Ishizaka, A., Labib, A. (2014). A hybrid and integrated approach to evaluate and prevent disasters. *Journal of the Operational Research Society*, 65 (10) 1475–1489.
- [24] Johanson, L. <http://lotsen.ivf.se/CBR/Shaping/index.jsp>, accessed 2018
- [25] Adla, A., Zarate, P., Soubie, J-L. (2011). A Proposal of ToolKit for GDSS Facilitators, *Group Decision and Negotiation (GDN)*, 1, p. 2011.
- [26] Adla, A. (2013). Modeling Cooperative Decision Support Systems with Hybrid Agents, *International Journal on Decision Systems Technology*, 5 (3) 31.48, 2013.
- [27] Nachet, B., Adla, A. (2014). An agent-based distributed collaborative decision support system, *Intelligent Decision Technologies (IJDSST)*, 8(1) 15.34, 2014.