Case-Based Reasoning for Diagnosis and Solution Planning[*]

Sima Soltani[†]

Supervisor: Dr. Patrick Martin[‡]

Technical Report No. 2013-611

School of computing
Queen's university

Kingston, Ontario, Canada

October 2013

[*] This technical report is based on the author's depth paper

[†] *soltani@cs.queensu.ca*

[‡] *martin@cs.queensu.ca*

Table of Contents

1. Introduction

Artificial intelligence (AI) has emerged as a field of computer science that aims to solve problems in a manner similar to that of humans. Although AI is still far from reaching human-level intelligence, there have been many successes in different areas and different applications such as computer science, medicine, finance, industry, transportation, communication, etc. [1].

The study of knowledge-based systems (KBS) (also known as expert systems) is one of the most successful branches of AI research. In knowledge-based systems, the model of knowledge must be elicited and implemented, often in the form of rules or objects, despite the depth of the domain knowledge that has to be covered [2]. Although model-based KBSs have been successful in many domains and applications [3] [4] [5] [6], several obstacles remain. The main difficulty is the elicitation of knowledge, mainly due to the requirement that the expert knowledge must be in the form of rules, which is not typically the way experts think about their domain problems and solutions. Other problems include the difficulty in KBS implementation for large scale systems, rule-based system's lack of memory, lack of robustness, and the difficulty of maintaining these systems [7] [2] [8]. The problems plaguing KBSs have been reduced by the emergence of case-based reasoning.

Case-based reasoning (CBR) is a problem solving methodology and a theory of reasoning that is based on the way humans think, reason, and solve problems in the real world [7] [9] [2] [8] [10] [11]. People tend to make decisions based on what they have experienced directly, or indirectly, through others' experiences. In the same way, a CBR system is an intelligent, problem solving system that reasons by first retrieving a relevant prior case from its memory of cases and then adapting solutions to prior cases (experience) to solve the new problem. Aamodt and Plaza [9] proposed a life cycle for case- based reasoning systems as shown in Figure 1. The main four steps of CBR, named as the four "RE"s, are retrieve, reuse, revise and retain [9] [10]. In the *retrieve* step, a new problem is compared to cases in the case library (case base) and one or more similar cases are retrieved. The solutions in the retrieved cases are *reused* for the new problem and the success is evaluated and noted. If the suggested solution does not satisfy the new problem, *revision* is required. The revised solution and its problem are retained in the case base for future use [9] [2][3][4].

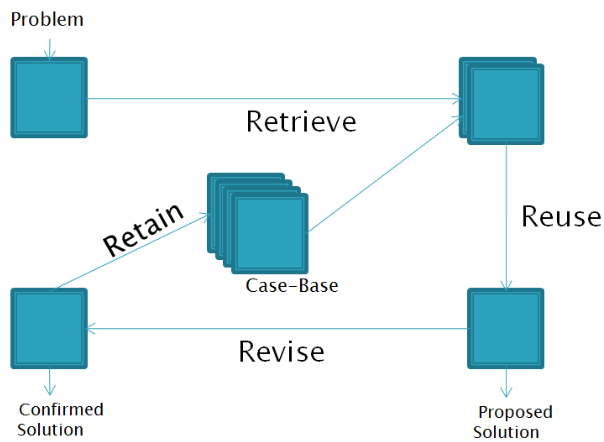

Figure 1:CBR cycle proposed by Aamodt and Plaza [9]

There is a vast number of applications using CBR from problem solving in applications involving design [2] [11] [12] [13], planning [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] and diagnosis [11] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47,48,49,50] to interpretive tasks like understanding and justification [2]. Recently, several web based applications have been built that use CBR [51] [52] [53] [54] [55].

In this paper we present an overview of CBR, and we survey two kinds of CBR applications, diagnosis and planning, and illustrate how CBR has been used in these applications. The remainder of the paper is structured as follows. Section 2 provides an overview of case representation. In Section 3 the different types of memory models used for case base representation are studied and their advantages and disadvantages are outlined. In Section 4 we survey the different methods used in the 4 "RE" steps of the CBR cycle in the literature. Section 5 focuses on the diagnosis and planning applications. We outline the methods that have been used and provide some insights as to which methods work better in these applications. Open problems in the field of case based reasoning and diagnosis and planning are discussed in Section 6. In Section 7, we summarize the paper.

2. Knowledge Representation in CBR

In order to use the previous experiences in the CBR cycle, cases must be represented in a structural manner. Several methods of representation can be used in case base reasoning and the decision of the representation method depends on the domain that the system is modeling and the types of similarity assessments and retrieval, which are chosen according to the requirements of the system [7] [8] [56].

The simplest format to represent the cases in the case base is to have simple feature-value vectors, which are good for cases with attributes of nominal or numeric. With this kind of representation, no relationships between the attributes in cases or relationships between the cases are shown and surface similarity-based retrievals can be used for retrieving from the case base [7] [8] [56].

In some domains, however, the attributes are complex, or there are some relationships between the attributes of a case, or relationships between the cases in the case base. Several representations can be used for cases according to the requirements of the system. Cases can be represented in the form of objects, predicates, semantic networks, scripts, frames, rules and concept maps [7] [8] [56]. In some domains with complex cases like planning and design, specific representations have been proposed and used [56]. Case bases also can be represented in the form of XML documents [57].

3. Case Memory Models

The case base should be organised in a manageable structure that can support efficient search and retrieval methods. Several case memory models (organisation) have been proposed that can be grouped as Flat memory model, Hierarchy memory models and network-based memory models.

3.1. Flat Memory [11] [58]

In a flat memory model, all the cases are organized at the same level. Retrieval time in this memory organization is very high since for each retrieval all the cases in the case base must be compared to the target case. Thus, for large case bases, this method is unacceptable [59]. Despite this disadvantage, the advantages of this approach which include maximum accuracy and easy retention, have lead to its use in many applications.

3.2. Hierarchy or Shared-Feature Network [59]

3-layer [60]: Cases in this model are structured in the format of a three layer network. The first layer nodes are the feature- values, the second layer contains the problems and the third layer is the solution layer. Separating problems from solutions makes it possible for different problems to share a solution and for a problem to have alternative solutions. The connections between the first layer and second one show the feature-values for each of the problems. The weights of the second set of connections represent how important a solution is for a problem case if it is a potential candidate solution. The restriction for this kind of memory model is that it assumes simple nominal and discrete numeric attributes and it cannot cover complex and continuous attributes. Figure 2 illustrates the three-layer structure of a case base.

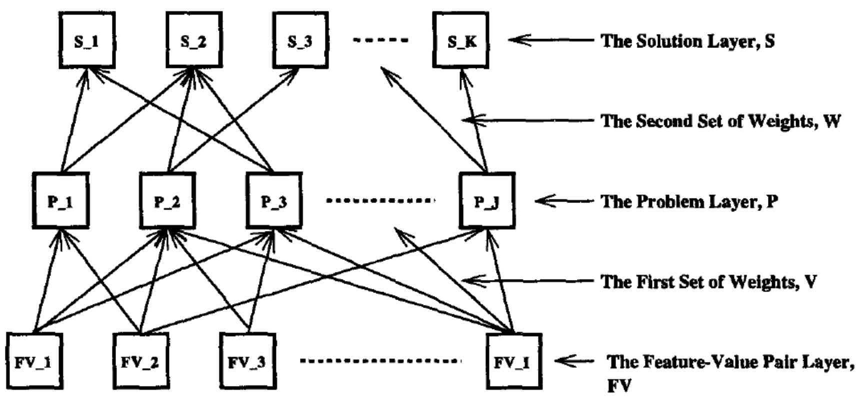

Figure 2: Three –layer structure of a case base [60]

Dynamic memory model [2] [9]: In this organization, which is also called Generalized Episodes (GEs), specific cases which share similar properties are organized under a more generalized structure (GE). Each GE contains three types of objects: norms which are features common to all cases under a GE, cases and index features that discriminate between a GE's cases. When a case is to be added to the case base, a search of the case base is performed and when a feature of the new case matches a feature of an existing case a new GE is created. Indexing the two cases under different indices discriminates these two cases below the generalized episode. Figure 3 shows the structure of dynamic memory model. The disadvantage of this organization is the explosive growth in the number of indexes with increased number of cases. A proposed solution is to limit the number of permissible indices to a limited vocabulary [2].

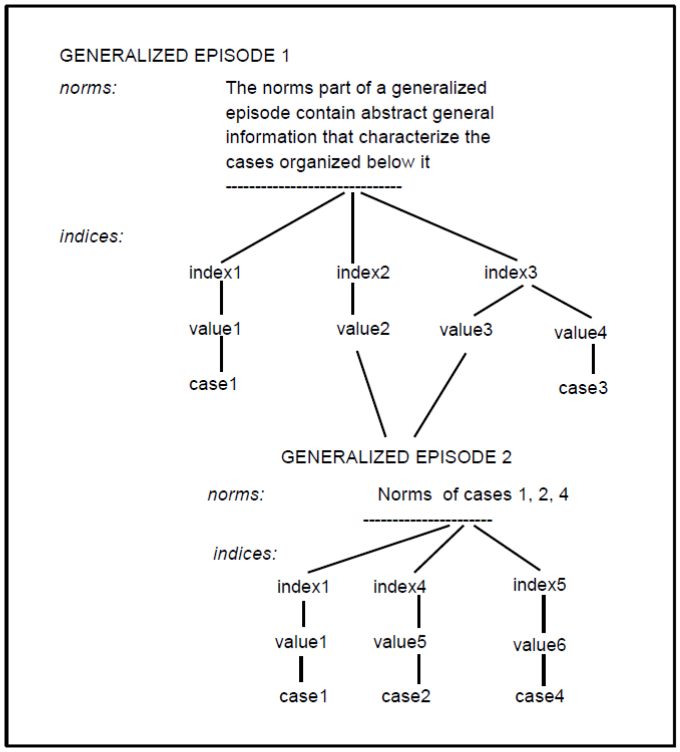

Figure 3. Structure of case base in dynamic memory model [9]

Hierarchy based on similarity between cases [61]: This organization is a k-d-tree (k- dimensional tree) that splits the case memory into groups of cases in such a way that each group contains cases that are similar to each other according to a given similarity measure. Figure 4 illustrates an example of a two dimensional search space and the corresponding k-d tree. This type of organization provides rapid retrieval; however, additions and deletions to the case base incur high maintenance costs due to the fact that the tree must be re-built for each update.

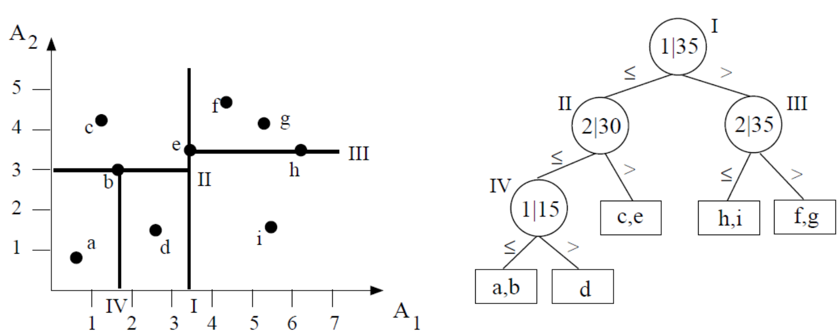

Figure 4: A two dimensional search space and how the k-d tree split the space [61]

Formal concept lattice [62]: This organization is based on formal concept analysis (FCA) which provides a way to identify groupings of objects with shared properties. In this method, the cases in the case base are organized as a formal concept lattice using their attributes. With the help of this lattice a set of dependency rules is discovered that can aid retrieval. At the time of retrieval, the description of the target problem is used to find the similar cases in the lattice. Using this method, different cases can have different attributes. An incomplete definition of the target case can also be handled.

Decision tree induction based models [58]: Decision trees partition the case base around nodes composed of single attributes. In making these trees, how much an attribute can discriminate the cases is calculated (e.g. with information gain of cases) and the attribute with highest discriminative power is located in the top of the tree. For the remaining attributes again the calculation will be done and the tree is made from top to bottom. This type of organization provides rapid retrieval, however, it does not handle complex attributes and suffers from complexity and high cost of maintenance [59].

Object- based model [63]: For representing complex domains, object oriented representation is an option. In this model, cases are represented as collection of objects and each object is described by a set of attribute-value pairs. The structure of objects is described by classes and they are arranged in a class hierarchy. There are two types of attributes for objects, simple types like integer or symbol, and relational attributes. The later type holds complete objects of a class. These types of attributes represent a directed binary relation, like a part-of relation, between the object that defines the relational attribute and the object to which it refers. These types of objects can represent complex cases.

Footprint memory model [64]: This memory model is based on competence. For construction of the model, the coverage and reachability of each case is calculated. Using these two measures, competence groups are formed. In each of the competence groups a sub-set of cases that covers all the cases in that group are selected as the footprint set of the group. Figure 5 illustrates the formation of competence groups and Footprint set in a case-base. The construction of this model is costly ($O(n^2)$) but the model's scalability is good and it is efficient for large case-bases.

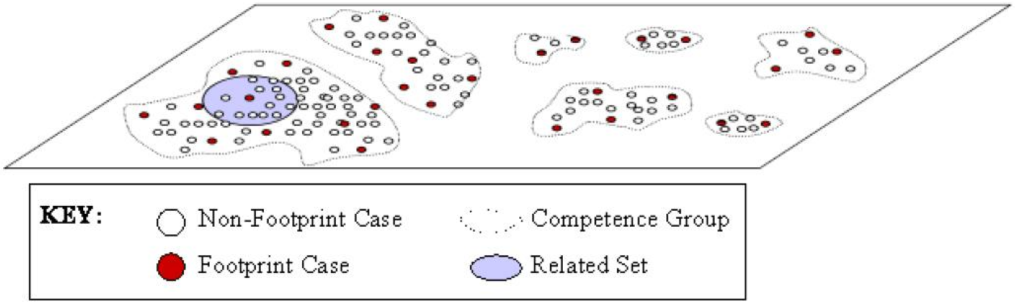

Figure 5: case base in Footprint model [64]

In general, in shared-feature networks it is difficult to maintain an optimal network as the case base expands [59].

3.3. Network Based Memory Models

Case retrieval nets [65]: The fundamental item in case retrieval nets is the information entity (IE). IEs represent any basic knowledge item, such as attribute-value pairs. A case consists of these IEs and the case base is a net with nodes for the IEs in the domain and additional nodes which are for cases. IE nodes may be connected by similarity arcs, and relevant arcs connect the case nodes to the IE nodes which make the case. Figure 6 illustrates an example of case retrieval net for travel agency. Construction of this organization is expensive, but these nets can handle partially specified queries. Using this kind of organization cases can have different attributes.

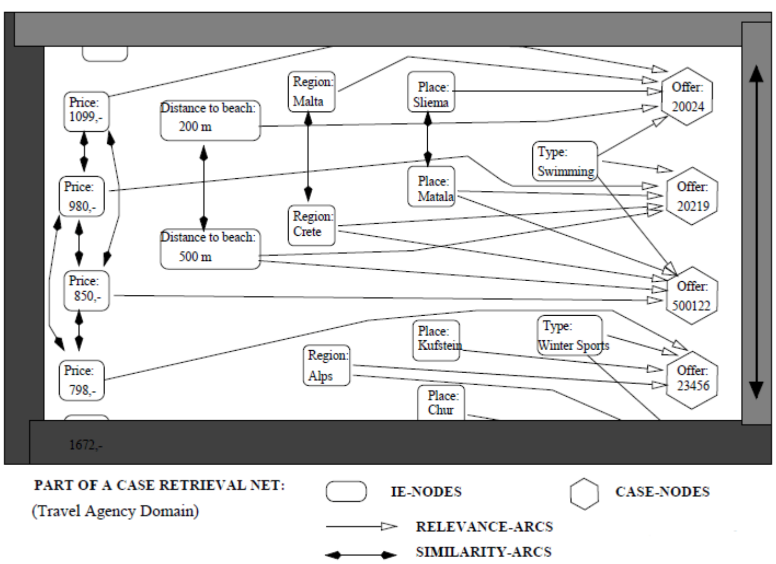

Figure 6 : Example of CRN in travel agency domain [65]

Category exemplar model [2]: The case memory in this model is a network structure of categories, semantic relations, cases and index pointers. This organization has three types of indices: *feature links* which point from problem features to a case or a category, *case links* that point from a category to its cases and *difference links* which point from categories to the neighbour cases where the differences to the current category are small. In this organization the categories are interlinked within a semantic network that represents a background of general domain knowledge which supports having explanation for some CBR tasks.

Fish & shrink polyhedral [66]: This model is proposed for domains in which the similarities between different cases are calculated just according to certain aspects, and they can be considered dissimilar when other aspects are regarded for comparison in another query, so similarities are dynamic. In this model, each case is represented in a polyhedral form, and each face of a polyhedral corresponds to one of the aspects. Case base is a network of these cases in which edges between each two cases show the similarities between them from different aspects. The label of each edge has a weight value that depends on the distance of the connected cases with respect to a certain aspect. Figure 7 shows an example of this model.

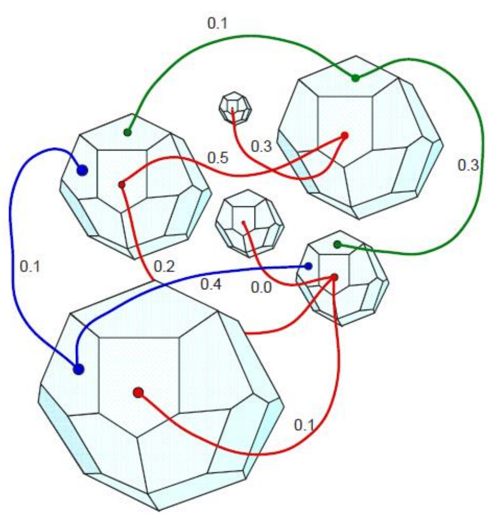

Figure 7 : polyhedral cases and case base as a network of these cases [66]

Figure 8 illustrates the hierarchy of different memory models. In table 1 advantages and disadvantages of memory models are shown.

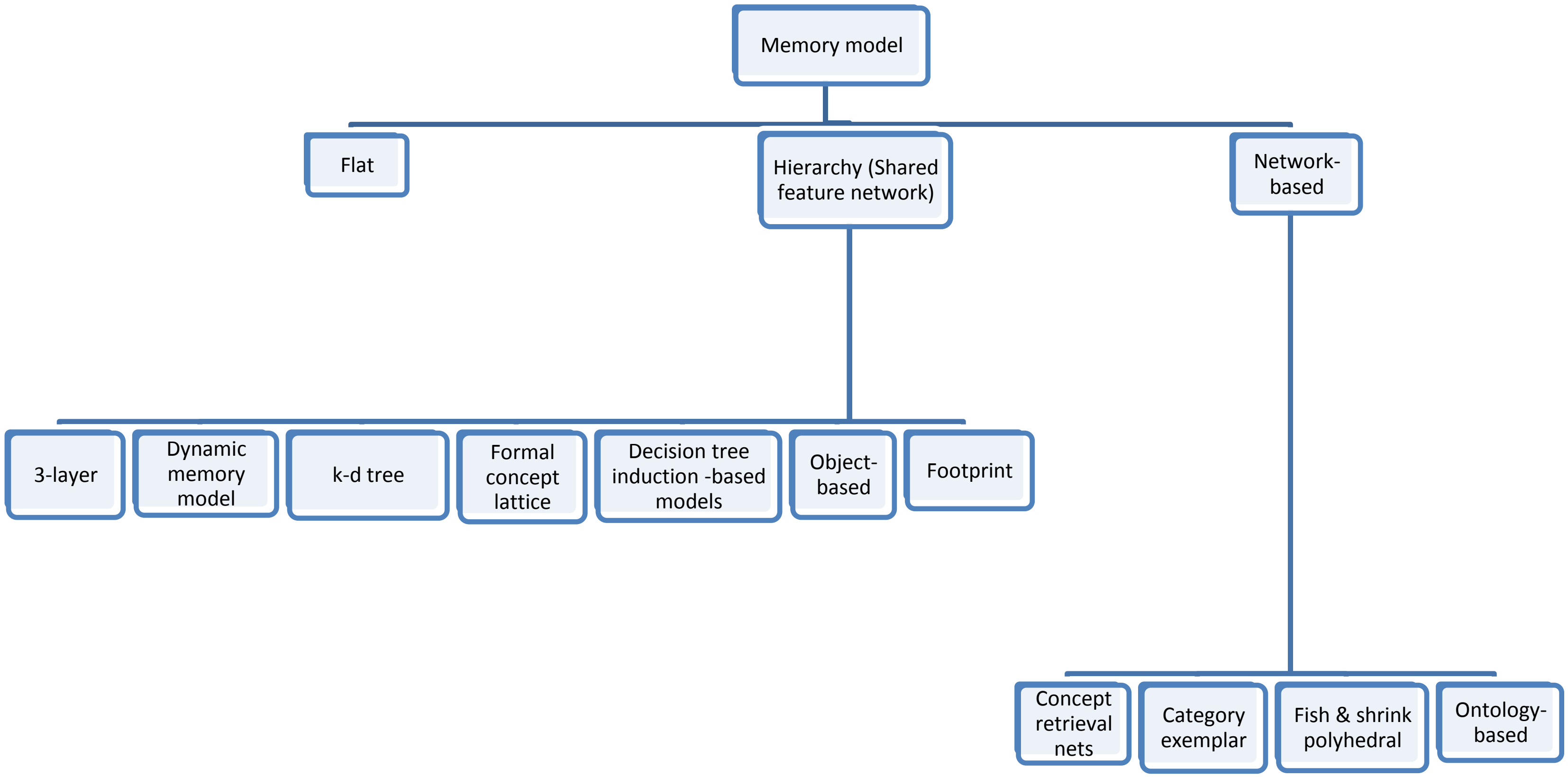

Figure 8 : Hierarchy of Memory models

Table 1: Memory models, advantages and disadvantages

Memory model	Advantages	Disadvantages
Flat Memory	- Easy retention - Maximum competence	Slow retrieval (Not applicable for large case bases)
3-Layer	- Supports different problem with same solution - Supports a problem with alternative solutions - Easy retention - Supports nominal attributes	Does not support complex attributes Does not support attributes with infinite values
Dynamic memory model	- Fast retrieval	- Explosive growth of number of indexes
k-d tree	- Rapid retrieval	- Does not support incomplete problem description - High maintenance cost - Does not support non-ordered attributes
Formal concept lattice	- Fast retrieval - Different cases can have different attributes - Can handle incomplete definition of target cases	- Supports just binary attributes
Decision tree induction based models	- Rapid retrieval - Supports non-ordered and nominal attributes	- High cost of maintenance - High complexity - Does not handle incomplete definition of target cases
Case retrieval nets	- Handle incomplete definition of target cases - Different cases can have different attributes - Supports non-ordered attributes - Supports complex attributes	- Costly construction
Category exemplar model	- Fast retrieval	- Complex structure - Need domain knowledge
Footprint model	- Fast retrieval - Handle incomplete description of target cases - Scale well and efficient for large case bases	- Costly construction
Fish and shrink	- Supports similarity from different aspects in different queries (dynamic similarity)	- Costly construction - Costly weighting of aspects between each two cases - Costly maintenance

	- Useful for complex attributes - Fast retrieval - Supports large case bases	
Object-based	- Fast retrieval - Can handle incomplete description of target cases - Supports complex attributes	- Need domain knowledge
Ontology-based	- Fast retrieval - Supports complex attributes	- Need domain knowledge

4. CBR Cycle

In 1994, Aamodt and Plaza [9] proposed a life cycle for CBR systems which is used by other CBR researchers as a framework. This cycle consists of four main parts; retrieve, reuse, revise and retain. Each of these parts includes a set of tasks and different methods have been proposed for each of them. In this section we give an overview on the tasks and the methods.

4.1. Retrieval

An important step in case base reasoning is the retrieval of previous cases that can be used to solve the target problem [11]. The input to the retrieval task is the problem description and the output is the cases that most closely match the new problem [9]. Among problem descriptors, more valuable features for retrieval have to be filtered. This filtering is done in a feature selection step. The cases in the case base are stored using these features and at the time of retrieval, these features are considered to be compared.

For retrieval of cases from case base in order to solve the new case, different retrieval techniques have been proposed which are described in this section.

4.1.1. Feature selection

Like other feature-based systems, in case base reasoning one area of research focus has been on how to select important features among all the features of the problem specification and weighting them to make the cases or to facilitate the retrieval. Different feature selection and assign weights to methods have been proposed in the literature. Guo et.al [67] proposed using Rough Set theory for reducing the number of the features in cases. Hsu and Huang [68] extract the features by evaluating the relevance between features and classes by a fuzzy measurement. They evaluate feature correlation, data appearance and gain ratio of the features to decide on the most useful ones. This type of feature selection is useful in classification tasks. Another method which was proposed by Smyth and Keane [69] and Craw [70] uses Genetic algorithms for learning both the features and their weights. Other feature selection methods used in case-based reasoning applications, but not specifically for CBR, are statistical methods like Fisher's criterion, t-test and logistic regression models [71].

4.1.2. Retrieval techniques

Given a description of a problem, a retrieval algorithm, using the indices in the case-memory, should retrieve the cases most similar to the current problem or situation [2] [8]. Every retrieval method is a combination of a similarity assessment procedure, which determines the similarity between a target case and a case in the case base, and a procedure for searching the case memory to find the most similar cases [64]. Factors that play major roles in determining the performance of a CBR system are the complexity and the accuracy of the case retrieval phase [59]. There are different retrieval methods in the literature classified based on the similarity assessment [72] which is shown in figure 9.

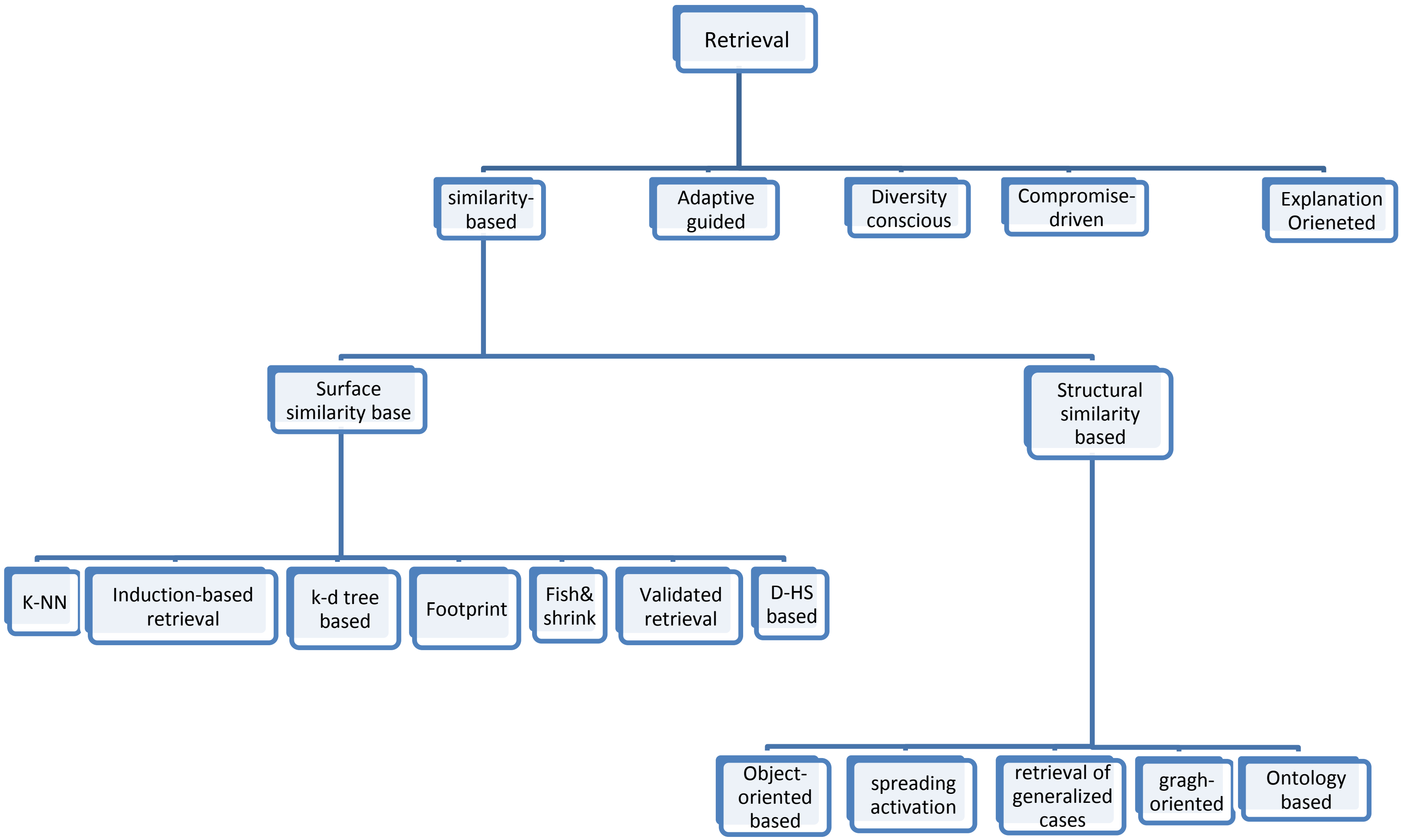

Figure 9: Hierarchy of retrieval methods

4.1.2.1.Similarity based retrieval

In this group of retrieval methods, the target is compared to the cases in the case base and relevant cases are found according to the similarity in the features or the structure of the cases. The features that are compared could be the surface features that are provided as a part of description of the cases, or derived features that can be obtained from the surface features and some inference from the domain knowledge [72].

Surface similarity based retrieval

i. Simple flat memory k-Nearest Neighbour retrieval (K-NN)

In this approach, the assessment of similarity is based on a weighted sum of features. Below is the typical equation for calculating the match between two cases:

$$\frac{\sum_{i=1}^{n} w_i \times Sim\ (f_i^I, f_i^R)}{\sum_{i=1}^{n} w_i} \qquad \text{Eq. 1}$$

Where w_i is the weight of feature i and the $Sim\ (f_i^I, f_i^R)$ function returns the similarity between the value of feature i of the input case and the retrieved case from case base [2].

The problem with this method is the retrieval time, which is $O(n)$, where n is the number of cases in the case base, so this method is not suitable for large case bases [2] [61]. However, the simplicity of this method leads to its use in other methods that reduce the size of the case base before starting retrieval. The main issue involved in searching in reduced case bases is the risk of missing the optimal cases since not every case is examined during retrieval [72].

ii. Induction methods

Induction methods are based on decision trees in machine learning (e.g. ID3 and C4.5). These algorithms determine the features that are best to discriminate between cases. Based on these features, a tree structure is formed to organize the cases in memory [2]. This group of methods cannot handle missing attribute values and are not suitable for case bases where the relevant importance of the individual case features change [64]. For retrieval, features of the target case are compared with nodes in the tree, until it gets to one of the leaves that contain similar cases.

iii. K-d tree based retrieval

K-d tree is a method for partitioning the data source using some hyperplanes. Every node in a k-d tree represents a subset of cases of the case base and the root of this tree contains all the cases. The partitioning attributes for building the tree are selected in a way that divides the case base into two equal size parts. Search for similar cases in k-d tree is done using recursive tree search. The average retrieval time for this method is log_2^n if the tree is optimally organized. This method cannot handle missing attribute values [61].

iv. Footprint retrieval

In this method proposed by Smyth and McKenna [64], first a competence model of the case base is created. A set of cases which cover all the cases in the case base are selected and named as footprint cases or reference set. For retrieval the search for the best case is done in two steps; first in the footprint cases (reference set) and after finding the best case in the reference set, search in the subset of case base which this reference case covers. For finding the best match, this retrieval uses the nearest neighbour approach. Figure 10 illustrate the two-step retrieval in Footprint retrieval method.

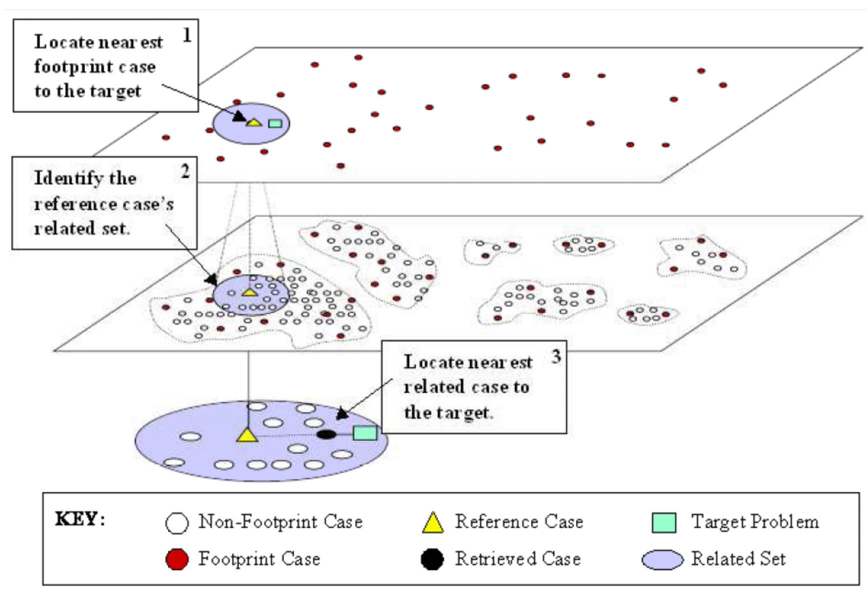

Figure 10: Two stage retrieval process in Footprint retrieval method [64]

In this method the number of cases that must be compared is very low in comparison to K-NN in flat memory while the quality (i.e. a function of distance between the target and the retrieved case) of the Footprint retrieval (FPR) is almost equal to K-NN. This method can handle missing attribute values.

v. Fish & shrink retrieval

This method of retrieval is used where the similarity of cases is dependent upon the aspect of interest. For instance, two cases could be considered similar with respect to a certain aspect, yet be considered dissimilar in another query based on a different aspect (Dynamic similarity). Also this method is useful where the cases have complex attributes.

In this method proposed by Schaaf [66], cases are linked according to specific aspect similarities. It is based on the assumption that if a case does not fit a query, then its neighbours also do not match, thus, leading to the elimination of many cases during retrieval.

This method supports large case bases and since similarity is evaluated according to some aspects, missing attribute values are not an issue. As mentioned in the paper, calculation of the similarities between cases for different aspects (weighting between each aspect of each two cases) is costly, although it is done at the time of case base construction, and can be viewed as pre-processing cost.

vi. Validated retrieval

Simoudis and Miller [73] proposed validated retrieval as a combination of simple retrieval with domain validation of the retrieved cases. In this method after a first stage of retrieval using simple retrievals (based on surface similarity based retrieval), further comparison is done between the target and the retrieved cases in order to reduce the final set of retrieved cases.

The problem with this retrieval method is the need for domain knowledge to build the validation model for the case base. Although the retrieved cases are the more accurate ones, the retrieval time for this method is more than just using the surface similarity based retrieval methods, rendering this method unacceptable for large scale case bases. Based on the retrieval method used for step one, it could be tolerant or not tolerant to missing values. This method is a good choice when the number of retrieved cases is more important than the retrieval time.

vii. D-HS (Discretized- Highest Similarity) retrieval methods

All the D-HS based methods use the cases in the case base as a training set to create a matrix where each cell $M(i,j)$ contains a list of cases whose normalized value x for attribute i lies in jth interval of the attribute. Figure 11 has an example of this representation for a case base with 3 features.

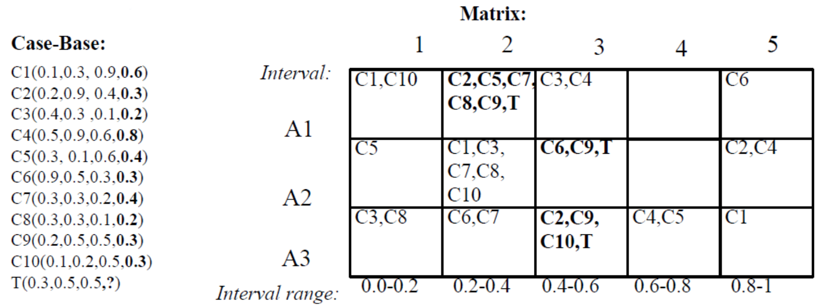

Figure 11: Matrix representation of the cases in the case base in D-HS based methods [74]

For continuous attribute values, they discretize the values into some intervals and for nominal attributes, the intervals are different values for that attribute. In retrieval, cases which have the most matching attribute values to the target case are retrieved. In D-HS method [74], using one of the difference calculation methods (e.g. Euclidean distance), the nearest cases among the retrieved ones in the first step are returned to the user. In D-HS-PSR (Discretised – Highest Similarity with Pattern Solution Reuse) [74] the retrieved sets are kept as a pattern tree in which each node has the attribute-value of one of the attributes and a pointer to the next attribute –value of the pattern. The leaves form the retrieval set. For large case-bases, there is a high probability that a number of different target cases have the same pattern tree.

Galushka and Patterson [75] focus on the issue of uneven distribution and propose D-HS^{E} which discretizes the attribute –values based on the entropy and the density of the values for each interval. Stephane et.al [76]focuse on the retrieval of the cases in discretized case bases using a query sphere algorithm in which the neighbourhood problem query consists of finding the relevant cases within a given distance from a given center location of the target problem. This method works better than previous methods for target problems that are near interval boundaries.

In general all the discretized retrieval methods work significantly faster than K-NN for large case-bases and they are domain independent [76]. They are also tolerant to missing values.

Structural similarity based retrieval

Although structural retrieval is computationally expensive because of the use of domain knowledge in formulating the structure, retrieval may find more relevant cases in comparison to similarity- based retrievals using surface features [72].

Several retrieval methods were proposed to retrieve relevant cases according to the structure of the case base. One of the first works on structural assessment was by Borner [77], where retrieval is done in two steps. In the first step relevant cases are retrieved from the case base using surface similarity assessment and then having some functions, system creates structural format of the target case and search for the cases with similar structures among the retrieved cases in previous step.

Below are some other techniques in structural-based retrieval:

i. Object-oriented based retrieval

One way to represent cases is in the form of objects where each of the attributes could be of simple types like integer or string or could be of type object. This forms a hierarchy of the object structure within which cases in the same classes of the hierarchy can be compared. The issue with this type of structure is when the target case and the case in the case base are not objects of the same class [72]. Bergman and Stahl [63] propose a method of similarity assessment between objects in different levels of the hierarchy. Using this type of retrieval, not all the cases are compared to the target case, so it is faster than K-NN. Also this method is tolerant to missing attributes. If values are missing for the target case, the higher part of the hierarchy is searched, resulting in more retrieved cases.

ii. Spreading activation method

In this method, the case base is organized as an interconnected network of nodes which makes the case attribute value combinations [64]. The spreading activation method was proposed by Lenz [65] and was customized by Aamodt [78]. The network representing the case base consist of feature- value nodes and case nodes which are interconnected to each other and the weight on each edge between nodes shows the relevance of two different node. For retrieval from this network, the features of the target case activate a set of the nodes in the network which in turn activate another set of nodes. If the activation has strength above a defined threshold, the activation spreads until some of the case nodes are activated. The strength of the activations depends on the weights that are assigned to the edges between nodes which can be learned automatically, or can be assigned by the experts. The problem with this method is the cost of construction of the network and weighting of the edges in the network which is difficult and time consuming. This method is efficient and flexible enough to handle incomplete case descriptions [72]. Parallel activation and spread of the activation signals in the network make the retrieval faster than k-NN.

iii. Retrieval of generalized cases

Generalized cases can be viewed as the implicit representation of a set of closely related point cases. Mougoui and Bergman [79] defined the similarity assessment as an optimization problem and do the retrieval of the cases by ranking the general cases so there would be no need to compare the target problem with all the cases. This work focussed only on attributes with real values where in Taratakovski et.al. [80] continue the work over mixed, discrete and continuous attributes. , this approach was applied to a real world application [81]. The main issue with this method is that generating the index structure can be time consuming, however, this is only need to be done once [80].

iv. Graph oriented retrieval

Graphs are commonly used for representing complex domains like planning and design. These graphs could be attribute graphs, semantic nets or conceptual graphs [82]. Different approaches to retrieval from graph structures have been proposed [68] [72]. Petrovic et.al. [82] propose a two stage retrieval that use a heuristic search called Tabu search. First a simple tabu search is used to rank all the cases in the case base by estimating the similarity degree. A subset of possible similar cases then will be present to the advanced tabu search and the most similar cases will be retrieved. The results of their experiments showed that in contrast to previous approaches, this retrieval method works for large case base in which the graph structure has several hundred vertices and there are several hundred cases in the case base. They used domain specific knowledge for a retrieval.

One of the forms of representing graph structures is feature terms. Arcas and Mantaras [83] propose a method named “Prospective” which does the retrieval by matching a partial description of the target problem with the patterns in the lattice of the feature terms which is the representation of the case base.

v. Ontology based retrieval

Ontologies can be used to make the case base where the cases are the instances of the ontology. Assali et.al. [84] propose a similarity computation that has two components: a concept base similarity which is dependent upon the location of concepts in the ontology and a slot based similarity which calculates the similarity of two objects based on the common attributes between them. They define a notion of similarity regions, which is a sub-branch of the ontology where concepts and instances can be compared. This eliminates the need to compare the target case with all the cases in the case base, therefore, making the method faster than K-NN retrieval.

4.1.2.2.Adaptive guided retrieval (AGR)

The effectiveness of a retrieval method is not just in finding the similar cases, but in identifying the useful cases [72]. In some applications, similar cases are not the ones that can be used in the reuse stage of the system, often because they are not adaptable for the target problem while certain cases with less similarity can be adapted so their solution can be used for the new problem. This issue in retrieval and its relation to reuse leads to studies on how to include the knowledge of adaptation in retrieval [85] [69]. In AGR, at the time of retrieval, matches between the specification features of the target case and a case in the case base are constructed if adaptation knowledge shows that the match can be supported during the adaptation stage. Also the similar cases are ranked according to the overall adaptation cost they would have. The experiments in these works showed that using this type of retrieval leads to less adaptation failure and less adaptation effort in the reuse stage [72] [69]. Retrieval cost for this method is more than simple K-NN. This method is tolerant of missing values.

4.1.2.3.Diversity conscious retrieval

In some systems like recommendation systems, the retrieval of cases deemed similar restricts the user's choices. In these systems, diversity plays an important role in the satisfaction of the customers of the system. To address this issue, Diversity –Conscious retrieval has been proposed [86] [87] [88] [89]. In this group of retrieval methods, the problem is how to make the trade off between similarity and diversity. Some examples of the algorithms that try to make this trade-off are greedy and bounded greedy algorithm [87]. Their experiments show that the retrieval cost is the problem with these algorithms.

4.1.2.4.Compromise-driven retrieval [90] and coverage optimised retrieval [91]

In recommendation systems, all the preferences of the user must be used at retrieval time. This is the reason that using K-NN cannot always return back the cases that satisfy the user. McSherry [90] [91] worked on how to increase the satisfaction of a recommender system's user by defining some preference criteria and retrieval of the cases in the case base that have the coverage over all the cases that could satisfy the user using these criteria. He included in his work an assumption, called a compromised assumption, that "if a given case C1 is more similar to the target query than C2, and differs from the target query in a subset of the attributes in which C2 differs from the target query, then C1 is more acceptable than C2." Although this method of retrieval increases the retrieval time, the retrieved cases are more acceptable from the user's point of view.

4.1.2.5.Explanation –oriented retrieval.

Explanation oriented retrieval explains how a question can discriminate between competing cases in recommendation systems. It can be used to explain the predicted outcome in classification and diagnosis systems, which could help in teaching the user about the domain. In planning, the explanation can be used to explain the plan failures in the system and to re-plan [92]. Some of the explanation based retrieval systems have been reviewed elsewhere [93] [92] [72] [94]. These systems need domain knowledge to make an explanation model (e.g. explanation tree) for the retrieval.

Although for most of the retrieval methods the type of application and needs of that application are the criteria to select a retrieval method, when more than one option for retrieval is available, the following aspects can help to decide [72] [65]:

- Efficiency of the method in both the speed and the efforts for searching in the case-base
- Quality of the solution with the measures like precision, recall and overall length of dialog with user and also how the method deals with the problems like noise, missing values or cases with different attributes.

4.2. Reuse

The second step in the case base reasoning cycle is reuse. After finding similar cases to the target problem, the system needs to reason according to the retrieved cases to find a reasonable and accurate solution for the problem. The reuse of the solution can be done in two ways. One is just copying the solution of the retrieved case as the solution for the target case (null adaptation) [2]. This is applicable to classification applications. However, in most applications, a retrieved solution cannot be used directly as the solution of the target case and some adaptation is necessary [9] [72].

4.2.1. Adaptation

Adaptation is particularly useful in constructive problem-solving tasks like design, planning and configuration. In these types of tasks, we do not have all the possible solutions in the case base, so by retrieving similar cases we find similar solutions and use the difference between the retrieved cases and the target case to modify the retrieved solution for the target problem [72].

Adaptation methods can be grouped as follows, according to how the changes on the retrieved solution could be achieved:

1. Transformational / structural adaptation: In this type of adaptation there exists domain-dependant knowledge in the form of a transformational operator {T} such that, applied to the old solution, it transforms this solution into a solution for the new case [9] [2]. Besides this knowledge, a control system is required to organize the operator [2]. Examples of this type of adaptation are parameter adjustment, abstraction and specialization, reinstantiation, model-based adaptation [2] [72] and the adaptation method proposed in Fuchs et.al. work [95].
2. Substitution adaptation: In this type of adaptation, the values appropriate for the new target problem are substituted from values in the old solutions [10]. Reinstantiation is an example of this kind of adaptation [2]. Craw et.al [13] also proposed a substitution adaptation in their work.
3. Compositional adaptation: In this type of adaptation, the adaptation takes parts of the solution from different cases that match corresponding parts of the user's input problem requirements. This adaptation can be guided by rules that consist of preconditions that check for equivalent parts of the problem description before copying parts of the desired solution [10]. An example of this method is proposed by Hanney and Keane [96] where the system searches for the rules that have all the differences between target and retrieved

case. If it doesn't find the matching rule, it tries to divide the differences into smaller parts and find matching rules for those parts.

4. Derivational adaptation: this type of adaptation, which is also called *derivational replay* or *generative adaptation*, looks at how the problems in the retrieved cases were solved. The cases hold the information about the method used for solving the retrieved problem including a justification of the operators used, sub-goals considered, alternatives generated, failed search path, etc [9] [10] [72].
5. Special purpose adaptation and repair: This adaptation method is for domain-specific and structure-modification that is not covered by transformational and substitution methods [10].

Extracting adaptation knowledge is a complex research issue. One method to learn adaptation knowledge is to make a training set from the case base by leave-one-out testing [13]. When one of the cases is removed from the case base, the other cases in the case base can be used to find the solution for the removed problem. The adaptation knowledge is saved as a case in the adaptation case base. Figure 12 shows an example of the cases in the adaptation case base.

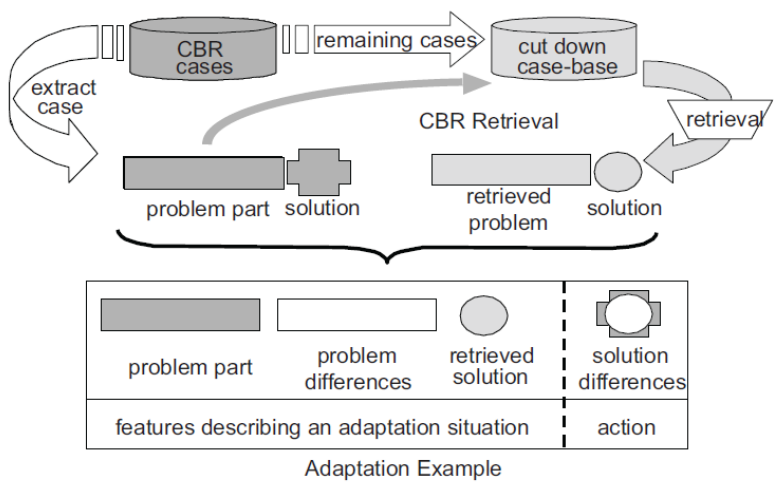

Figure 12: Learning adaptation data from case base [13]

Inductive learning of the adaptation knowledge is another method named as AKA [29]). In this method, a similarity path between the target and the retrieved case is calculated in the form of a similarity path and then for each step in the path, adaptation knowledge is extracted from domain knowledge and is kept in the adaptation knowledge base. Finding adaptation rules by comparing each pair of cases and the differences in their solution is another method which is proposed in Li et.al. work [97]. Case base mining is also used to extract adaptation knowledge from case base [98] [96].

4.3. Revision

After choosing to reuse a solution from the retrieved cases for a new problem, it may be discovered that this solution is, in fact, incorrect, thus providing an opportunity to learn from failure. In this phase, which is called *revision*, the case solution is evaluated and if the solution is incorrect, then domain specific knowledge is required to repair it [30]. CHEF [30] was one of the CBR systems that included revision. In CHEF, causal knowledge is used to generate an explanation as to why the solution does not fit the goals of the system. These explanations are used to modify the solution [7]. Another method of correcting the proposed solution used in CASEY [11] and the proposed system by Pertinale et.al is to use a model based system (i.e. a domain theory implemented with rules) that at the time of failure finds the correct solution for the problem. MBR should be a complete model of the system. This method is proposed for systems where complete information about the domain is accessible for making the model for the domain. Using a case base reasoning system renders the task faster and allows it to work online.

In general, revision can be viewed as two tasks: diagnosis of failure and solution repair [30]. For diagnosis, one of the following ways can be used: 1) execution of the solution and evaluating the outcome 2) using a simulation -model of the real world and evaluate the solution using the model. This solution is safer and more cost effective. 3) Experts also can help in diagnosing the failure in solutions. The expert evaluates the solution using his/her experiences. 4) Use the case base itself to identify the failure. In this case, to assist in problem diagnosis, in addition to the specification of the problem and the solution for each case, knowledge about the conditions under which the failure may occur must also be stored in the case base [99].

4.4. Retention

In a case base reasoning system learning is done in the retention step. In this step, the new case will be added to the case base according to some policies in the system. Retention includes adding knowledge and new cases to the case base, all which needs to be indexed, as well as deleting cases from the case base in order to restrict its growth. Having new information about the cases in the case base and the knowledge system obtained in the previous steps in the cycle, indexing of the case base and other knowledge would be changed in this step. Different retention (maintenance) strategies fall into one of two groups: maintenance of the content of the case base and maintenance of the organization of the case base. Research on maintaining the content involves work on the reduction of the case base and the deletion and addition policies. Maintenance of the organization is related to indexing the case base in order to make the case retrieval faster and more efficient. Another type of maintenance is the maintenance of feature weights [60] [100] which is not included in this paper.

4.4.1. Case base Size Reduction

The maintenance of the case base content is important for two main reasons: 1) to control the size of the case base and reduce the retrieval time. 2) To eliminate useless cases and irrelevant and redundant instances that render the case base inconsistent in order to increase the accuracy of the CBR system [101].

In early knowledge based systems, there was a belief that more knowledge is a good thing. As knowledge based systems become more practical, it has become obvious that there is some "harmful" knowledge that can degrade the performance of the system. In case based reasoning, too many cases stored in the case base can lead to expensive searches. This is termed the *Swamping Problem*. As the number of cases in the case base grows, the expense for searching for similar cases increases [102] [72].

In early works on case base maintenance, the deletion policies were just random deletion or selective deletion according to the performance benefits of the cases in the case base [102]. Smyth and Kean [102] define two concepts, coverage and reachability. Coverage of a case is the set of cases that it can be used to solve. Reachability of a case is the set of cases that can be used to solve the case. Based on these measures, they group the cases accordingly into four groups: Pivotal, Auxiliary, Spanning and Support. Smyth and Kean propose a competence model that is used in their deletion policy. First they propose the Footprint deletion (FD) policy where deletion is done according to the competence. The problem with this method is the possibility of preserving low utility cases while deleting high utility cases. To solve this problem, they proposed Footprint Utility Deletion (FUD) where the decision to delete a case is based on both performance and competence. They continued their work by proposing a case selection method based on the coverage of the cases in the case base [103]. Continuing the work of Smyth and Kean, Lu et.al. [104] proposed a new competence model and defined steps to be taken after the deletion of a case to preserve the efficiency of the model for retrieval. Haouchine et.al [105] expand the deletion policy proposed by Smyth and Kean by defining two types of spanning cases (inter-spanning and intra-spanning). In their work, they keep all the Inter-class spanning cases and remove all the Intra-class spanning cases except the cases which have coverage less than a predefined threshold. According to their experiments, they had positive results in terms of case base reduction, accuracy and competence.

Leake et.al. in their work in 2001 [106] argue that case selection based on coverage as proposed by Smyth and Kean [103] is not a good criteria and as an alternative, they propose performance-based metrics for case selection. The performance is calculated according to the adaptability of the cases in the case base. For each case they calculate a metric, named relative adaptation performance, which is the percent savings the case provides compared to the worst alternative case that solves the problem. Experimentally they proved that their method works better than previous methods specially in non-uniform case distribution environments when some regions in the case base are used more than other regions.

Zhu and Yang [107] prove that using FD and FUD policies, the case base may suffer reduced competency after deletion. They propose an additional policy for case base reduction where a new case base is made from the original case base by selecting K cases with the highest coverage (K is the defined size for new case base). The problem with this policy and FD and FUD is the time complexity of maintenance ($O(n^2)$) which is a high cost for the CBR system [108]. Another case reduction technique based on addition is JUST [109] which selects the cases from the original case base and adds them to the new case base using some justification criteria such as the size of the new case base and the minimum accuracy to terminate the addition of cases to the new case base. This system is just for classification tasks. Ni et.al. [110] also propose an addition technique by outlier mining and sieving cases to formulate a new case base from the most valuable cases. The values of the cases are calculated using a goodness measurement which is based on coverage of the cases. Their algorithm also has a complexity of $O(n^2)$ which is costly for the CBR system.

Lawanna and Daengdej [108] proposed a method called DRCBM which does the deletion in such a way that it maintains the competence of the case base maximum. In their algorithm, they also have indexing on features that have maximal coverage and minimal reachability between cases. In their evaluation they compare their method with FD & FUD and case addition in [107] and prove that their method is more efficient since it achieves a better case reduction rate with a finer competence reduction. Also they showed that their method has a higher reduction rate in comparison to the other two methods.

Another work on case base retention is Adaptive Case Base Reasoning [111]. In this system a case base is formed by retaining and forgetting cases. Different retention and forgetting strategies were used in the system and a measure called “goodness measure” was used to decide on the cases to be added or forgotten. The goodness measure is calculated using reinforcement learning. At the time each case retrieved has the correct solution for the new problem the goodness of the case increase and when it has wrong solution the goodness decrease. A simple calculation for goodness was shown to not affect the efficiency of the system and their system generated a more compact case base in comparison to other CBR maintenance methods.

The deletion policy used by Romdhane and Lamontagne [112] was based on the usage of the cases in case retrieval and the reinforcement value of each case. According to their experiments case usage is a good criterion on which to base the decision to delete cases, but reinforcement value just can be a contribution when it is combined with other criteria.

Rough set is another method used in deletion policies. Salamo and Golobardes [101] [113] proposed two deletion policies using the foundation of rough set theory. The proposed methods include Accuracy-Classification Case Memory (ACCM), Negative Accuracy-Classification Case Memory (NACCM), SortOut case memory (SO), SortOut Internal Case Memory (SOI), SortOut Mean Case Memory (SOM) and SortOut Mean Internal Case Memory (SOMI). They propose a new definition of coverage and reachability of the cases using rough set theory. ACCM keeps all the cases that are near the outliers and maintain all the internal cases in a way that covers all the internal cases. In NACCM, selecting cases starts from internal cases and then continues for outlier ones. Sortout case memory policies are based on grouping the cases in coverage groups and their difference is on the number of cases that have to be deleted from each coverage group. The case in the coverage group which has the maximum coverage is called the master case. In the SO method, for each of the coverage groups just a case with maximum coverage is kept. SOI deletes all the cases except master case if it can solve all the cases in the coverage group correctly, otherwise all the cases are kept in the case base. The next two methods (SOM and SOMI) try to make a new case base from the original one instead of deleting the cases from the case base. The reduction of cases obtained from these methods was not as large as previous algorithms [109].

Neural networks and fuzzy logic techniques were also used in reducing the case base. In Shiu et.al [114] the system uses a neural network as a classification tool to divide a case base into various classes and each of the cases has a fuzzy membership to each of the classes. After classifying the cases, the coverage of all the cases is computed and the cases with highest coverage are selected for addition to the new case base. The problem with this method, like other addition methods (making a new case base from the original case base) lies with adding a new case to the case base. After a period of using the system, if just the new case base is used (and new cases not added to the case base) then it lose its efficiency. Also, rebuilding the case base is costly. How to overcome this problem is an open problem in maintenance. Yang and Wu Work [115] also use the same clustering method, however, retrieval is based on the information gain of the features. The features are presented to the user, and based on the values the users assigned to each feature one of the clusters are selected and the cases contained in the cluster will be returned to the user.

Harmful cases need to be eliminated from a case base. One type of harmful case is noise cases (cases that contain errors in values used to represent the case) which can decrease the efficiency of CBR system by returning incorrect solutions for the target problem [116]. Another type of harmful case is boundary cases (especially for classification applications), that is, those located near the boundary of a class. In Massie et.al [116], a ratio is calculated which can provide the potential harmfulness of a case in the case base. The ratio gives an indicator of the positioning of the case in relation to the cases which have the same classes and the ones with different classes within the case's local neighbourhood. According to this ratio, they find noisy and harmful cases. The deletion of the cases is based on the threshold specified for the ratio, and this threshold is domain specific. The proposed policy, Threshold Error Reduction, increased the accuracy in many applications.

Inconsistency in the case base is another reason for the need for maintenance of the case base. Racine and Yang [117] propose the use of a rule base system for finding inconsistencies in the case base. The main problem with this method is the knowledge acquisition for the rule base system.

Portinale et.al [118] propose a case memory management schema with the idea that when a case represents for learning (retention), the cases in the case base that cover the same portion of the problem space are consider to be replaced by the new one. In their later work [119] they propose a failure-driven deletion method which is also called learning by failure with forgetting (LFF). The main idea of their policy is to find the false positive cases during the usage of the case base and delete them. Also their system distinguishes another group of cases which are old cases. In a specified time interval, the system detects the old cases and deletes them.

The following table have a summary on different methods of case base size reduction, and their pros and cons.

Table 2. Case base reduction methods

Method	Criteria	Pros and cons
Footprint Deletion	Competence	High time complexity
Footprint Utility Deletion	Performance and Competence	High time complexity
Hauchine [105] (expands smith and Kean work)	Competence	High time complexity
Leake et.al [106]	Performance (performance according to adaptability)	Good for non-uniform case distribution
Zhu and Yang [107]	Competence	High time complexity
JUST [109]	• Size of new Case base • Minimum accuracy for termination	Just for classification tasks
NI AT.AL [110]	• Outlier cases • Goodness (coverage)	High time complexity
DRCBM [108]	Competence	Better reduction rate in comparison to FD &FUD
Adaptive CBR [111]	Goodness measure (reinformancement learning)	• Low time complexity • More compact CB with same efficiency
Salamo and Golobardes [101] [113] (ACCM, NACCM, SO, SOI, SOIM)	Competence using rough set theory	
Shiu et.al [114]	Coverage (using fuzzy logic and neural networks)	• Problem of adding new cases • Costly rebuilding case base
Threshold Error Reduction [116]	Harmfulness ration	• Elimination of harmful cases • Just for classification applications
Recine and Yang [117]	consistency (intra-case and inter –case)	• Elimination of inconsistent cases
Failure-driven deletion (LFF)	• Oldness of cases • False positive cases	• Elimination of harmful cases

4.4.2. Indexing

A part of the retention step is indexing, and it constitutes one of the main issues for efficient retrieval of cases [59]. Indexing the cases in the case base was applied as a method for improving such efficiency to combat the effects of the utility problem. It helps to reduce the search time and increases the efficiency of identifying a possible solution by means of making only a selective portion of the case-base available [120].

According to Watson [8] [2]indices should: 1) be predictive, 2) address the purpose for which the case are used, 3) be abstract enough to allow for widening the future use of the case base and, 4) be concrete enough to be recognized in the future. We can find different types of indexing in the literature which can be categorized into the following groups:

1. *Difference-based techniques*: This type of indexing selects features that differentiate a case from other cases as indices. A sample of using this technique is CYRUS [2].
2. *Inductive learning methods*: These methods, which are based on inductive learning in machine learning, identify predictive features and use them as indices. Different types of tree-base inductive learning methods fall into this group [2] [114]. Genetic algorithm-based methods and neural network based methods are also included in this group [114]. These methods require a certain number of training examples [121] [93]and they perform poorly when insufficient data is available. Another problem with this group of indexing techniques is that the learning phase is complicated because of the complex architecture which is used. Also maintenance in these techniques is difficult because after adding new cases or deleting some cases from the case base, the indices need to be recalculated [59].
3. *Explanation –based techniques*: In this group of indexing methods, relevant features for each case is determined and features that are predictive according to the relation between features and the explanation on each case in the form of domain knowledge will be selected as indices [93] [122]. Ontology-based techniques [123] could also be classified in this group. In these techniques, domain knowledge (the causal model) is required [93] [122].
4. *Similarity –based generalization*: In these techniques, indices will be created in two levels, one level is for the abstract cases, which are the cases that share some common features, to differentiate between different abstract cases, and unshared features are used as indices for original cases [2] [59] [124] [64] [125] [126].

5. *Discretised-based techniques*: These methods discretise the feature space and the indexing space will be the discretised feature space [127] [120] [74] [75] [76].
6. *Dynamic indexing*: In this method, indexes will be made online at the time of a new input to the system. This dynamic indexing is based on the weighting of the features which will be set by the domain expert. The level rank and the query for retrieval will be made according to these indices [128]. Disadvantages of dynamic indexing: 1) requires specific domain knowledge from an expert. 2) this indexing method cannot cope with the addition of new cases to the case library. This is due to the fact that the risk level of the attribute – value has to be recalculated each time we want to add a new case [120].
7. *Bitwise indexing*: This method of indexing works on discrete features with finite values. According to feature-value pairs for each case, a bit string shows the case and the comparison is between these bit strings at the time of retrieval [127].
8. *Introspective indexing*: In this method, introspective learning is used to permit the CBR system to detect the features that are implicit in the original problem and not explicit in CBR system's initial indexing schema and set them as indices in order to direct retrieval towards cases that can easily be adapted [129].

5. Diagnosis and Planning with Case Based Reasoning

5.1. Diagnosis and case-based reasoning

Diagnosis is the identification of the nature and cause of a problem [130]. Diagnosis can be done by exploring the exposed symptoms, the system state, the general specification of the system and the operating environment. In diagnosis, the behaviour of an observed system is checked for previously defined problem conditions to explain the current problem that the system is experiencing. Reusing previous experiences in diagnosis can result in faults being corrected more quickly and more consistently and is the method employed by CBR [131].

Several diagnostic applications have been developed since the introduction of CBR systems. The most dominant area that has used CBR as a technology for diagnosis and fault detection is the medical field [71] [49] [43] [37] [44]. CBR is also used in industrial applications [45] [46] [47] [38] [41] [50], as well as software fault detection and troubleshooting systems [33] [36]. PC, computer network, printer troubleshooting and database monitoring [31] [48] [35] [42] [132] are other examples of the use of CBR for diagnosing problems. In the next few paragraphs we have an overview on the methods that diagnosis CBR systems used for case representation, case base organization and different stages of CBR.

Different representations can be used in a diagnosis application. The most common representation is feature-value representation [33] [37] [38] [40] [43]. Semantic networks have also been used in diagnosis applications case representation [35] [31]. Objects are used by Wang and Hsu [42]as a representation for cases in the case base. Besides these general representations which were used for cases, in some of the applications, specific representations were proposed. Lopez and Plaza in their system [20], a planning system for diagnosis, proposed a sequence of decision steps or episodes for case representation. Constraint net [35], proposed by Lee and Ng, is another specific representation for the cases. The case representation that is used in different diagnosis application is based on the domain of the application and the requirements of that domain.

The dominant memory organization in diagnosis applications is the hierarchical memory model [48] [20] [42] [34] [32]. Semantic net is the second most popular memory model used in diagnosis applications [31] [35] [19]. Flat memory model is rarely used in these applications [33] because of its disadvantage on retrieval time.

For case retrieval in the CBR systems, K-nearest neighbour is the most frequently used retrieval technique [33] [37] [38] [40] [43]. For applications that have a type of semantic network, spread activation is the retrieval method which was used [20] [31]. For applications using constraint net for the representation, retrieval is done by finding a partial match between the constraint net of the target problem and the constraint nets of the cases in the case base [35].

In the reuse phase of the CBR system, the diagnosis applications mostly directly use the solutions of the retrieved cases from the case base because the diagnoses systems are the classification systems that just classify the target problem in one of the classes of diagnosis. When the retrieval step retrieves more than one case and these cases contain different diagnosis, typically a vote is carried out to decide between the possible solutions and the solution with highest vote is returned as the final solution or diagnosis [38] [43].

The solution forwarded from the reuse stage mostly is evaluated by an expert according to the different diagnosis applications reported. The system implemented by Melchiors and Tarouco [31] used a validation model to help the expert to evaluate the proposed solution.

Since the sizes of the applications reported in the literature are small the addition of new cases were used in the retention step and no case based reduction were mentioned. In these application, new cases were added to the case base if they were different from the cases in the case base, or if the diagnosis of the system was incorrect for the target problem [43] [35] [34].

In some of the diagnosis systems the treatment of the diagnosed problem also have been included in the system either as a part of each case [48] [35] [19] [36] or as another CBR system which the input of it is the output of the diagnosis CBR system [33]. Another method of having treatment for the diagnosis system could be using a planning system for planning how to treat the diagnosed problem.

5.2. Case-based Planning

After diagnosing the problem, a plan to solve the problem must be devised. Planning is a search problem to find a sequence of actions that can transform an initial state of the world to the given goal. Therefore after diagnosis, the initial state is the problem state of the system and the goal state is the state that the system revealed from the problem. Planning is also done using CBR by reusing past successful plans in order to devise plans for new, similar situations [99] [17] [14] [28] [16] [29].

Case based planning (CBP) was first proposed by Hammond [30] in CHEF. In his work he explained the expectations from each part of a case based planning system. CBP is used in different domains like medicine [20] [19] [17] [29], industry and manufacturing [18] [14], cooking [30], disaster management [15] [21], RoboCup games [16], real time strategy games [27] [22], agent based service- oriented systems [26] [24], multiagent systems [25] and route planning [129].

5.2.1. Retrieval and organization of the case-base

In the representation of the plan cases, the problem features are the initial state of the system and the goal. For the solution part, the plan for the problem is kept. Other knowledge, in addition to the initial state, goal and the final plan, may be maintained in cases for CBP systems. For instance, what failure the plan can avoid [30] (A very simple example is that if the goal is to go outside, then a plan to go outside with an umbrella avoids the problem of the planner getting wet). Based on the adaptation that the system uses, the intermediate steps to the final plan and the consequences of each step may be kept as knowledge related to the case. The representation of the cases in the case base for CBP is an issue because of the relationships between different parts of knowledge for a case and the effect of this representation on the efficiency of retrieval [133] [134]. The case memory model for CBP is usually hierarchical [18] [20] [24] [26] or network based [19].

Depending on the case memory and case organization, different retrieval methods were used for case based planning. Bergmann et.al. [134] discuss that adaptation guided retrieval or a hybrid method combining adaptation guided retrieval with other methods is most suitable for case-based planning since adaptation is an important part in these systems and retrieval should select adaptable cases.

5.2.2. Reusing previous solutions

A crucial step in case-based planning is the reuse phase, which requires strong plan adaptation capabilities [56]. The two common adaptation methods that are used in case-based planning are transformational and derivational adaptation [133] [135] [134] [99]. Derivational adaptation has advantages over transformational adaptation approaches for planning. It is more flexible, because the planner replays the derivational trace with the new problem, and also there is no requirement for predefined transformational operator [136]. This method, however, requires that additional information about successful and/or failed planning decisions be recorded as part of the case representation [56] and, as previously mentioned the traces of planning steps for each case raise some issues with case base organization of the representations of cases.

In CBP it could be the case that using different plan cases instead of just one similar plan case results in a more efficient solution for the problem. Different methods were proposed for merging different plan parts like Constraint-based CBP [23], sequential retrieval [24] [26] and using Demspter-Shafer theory to fuse the retrieved plans [17].

A detailed review of different adaptation methods used in CBP has been done by Avila and Cox in 2008 [136]. In their review, they analysed the research on adaptation types for planning, role of cases in planning adaptation, the knowledge to be kept for different type of adaptation, how adaptation works when more than one case needs to be used for solving a target case, representations of the plan cases and finally the complexity of plan adaptation.

5.2.3. Revision of Solutions

Different applications use different methods of evaluating the proposed plans. Expert evaluation is the method used in most applications [17] [21] [20] [18] [22]. In some of these applications a simulation model is used in addition to the expert to evaluate the proposed planning [18] [21]. Using the real world as the evaluation method is rarely used in case based planning due to the high cost of failure in the real world. ROBBIE proposed by Fox and Leake [129] is an example of a CBP application that used real world evaluation of the plans. For repairing the failures, three methods were used in CBP applications: 1) the expert changed the plan according to the problems in the proposed plan, 2) the system, using a model of the domain, or using some pre-defined rules, changes the plan [27] [21] [129] [30], or 3) a combination of these two methods [18] [22].

5.2.4. Retention of new cases

Indexing in CBP applications are mostly on the goals of the plan and the failure it can avoid, as proposed first by Hammond [30]. In some applications like TOLTEC [18], indexing has been done on the constraints of the plan. For maintenance of the case base, methods used in the literature for CBP were based on the usage of the cases and the inference efficiency of the cases [14].

One of the issues that researchers have explored is the complexity of case base planning [133] [135]. In some works, by formalizing case base planning, especially the adaptation phase of the system, they proved that adaptation of the plans using derivational adaptation is as complex as planning from scratch.

6. Open Problems

The knowledge elicitation bottleneck is still a problem in CBR. Those works that make the assumption that this is not a problem for CBR are modeling within a domain where complete information is available. However, this is rarely the case in real world applications. Knowledge elicitation in CBR is required not only for the cases themselves, but also for the selection of the features for each of the cases. In addition, there is a need for elicitation of knowledge about similarity assessments for each of the features and also for the overall case at the time of matching. Knowledge about how to adapt the previous solutions to the new problem is also knowledge that must be elicited. One open problem, then, is how to obtain all the required knowledge to formulate an effective case-base.

Another open problem in the field of CBR is how to deal with symbolic attributes and attributes with continuous values. Most of the methods for retrieval make the assumption that the attributes have discrete numerical values. The problem with features with continuous values is decision on the boundaries at the time of comparing and matching the features. This is important at the time when some indexing is done on these features to reduce the number of comparison in retrieval. Elicitation of the knowledge for adapting the symbolic attributes and the similarity assessment for this type of attributes still requires more research.

In some domains like fault diagnosis, the cases are heterogeneous (different cases have different features). How to deal with this heterogeneity for CBR different steps is still an open problem.

In the retention part of the CBR cycle generally deletion, addition and editing of the case base is still an open problem. Retention estimation of a case base and identifying missing or redundant cases are important areas of research. Maintenance of the case base is done to decrease the utility problem, but itself adds a burden to the system. It has to be examined if the maintenance always increases the performance of the overall CBR system while decreasing the utility problem of the system.

The Web is a rich source of data and information, and learning the cases and the rules or knowledge from the Web for reuse and revision is future work in the field of CBR. Text case based reasoning is the focus of much research these days. Since most of the resources on the Web are in unstructured format or text format, how to retrieve, reuse, revise and retain cases from these resources is ongoing research. An example of these applications is when web pages like F&Qs are considered as knowledge resources.

In most of the studies on CBR there is the assumption that revision will be done manually by a domain expert. There are few studies that focus on learning revision rules, and this part needs more research.

Many systems change over the time so how to detect and integrate these changes is a topic of ongoing research. The maintenance of the case base, indexes and updating of reuse and revision knowledge according to these changes over time need more research. In general, research on different parts of CBR cycle (retrieval, reuse, revision and retention) is still ongoing.

Providing CBR systems as a service could also be a focus of research. Factors like security, distribution, scalability and management are important for a case base service. A service provider can provide both private and public case bases in the domain. Management of the case bases, including how to provide different types of memory organization, retrieval, reuse, revision and retention methods and interfaces according to the user application are open problems. When offering CBR as a service, the scalability of the case base system is an important factor. In addition to the common methods for dealing with huge case bases (e.g. case reduction and maintenance of indexing), management of available resources and the user's required performance should be considered. Distribution is a method for dealing with huge case bases and having better performance, but in this method also there are some challenges like how to distribute the cases, the retrieval methods for distributed case bases in order to have the best performance and having multi agent system for processing.

7. Discussion

Since 1976, rule-based reasoning is applied in many computer-based diagnosis applications (like medical decision making) [137], but there were several problems in applying this method in broad and complex diagnosis applications. One of the requirements for implementing a rule-based system is that the problem domain should be well understood and be constant over time. Also the domain theory should be strong and the knowledge of the domain expert should be extracted in the form of rules which is hard. Experts do not think about the domain problems and solutions in the form of rules, which leads to the absence of general rules for the domain.

On the other hand, CBR is known to be good for domains that are not completely understood and knowledge is incomplete. Also since CBR learns automatically as the system is used, it is very good for domains where the background knowledge is insufficient at the time of implementation but evolves over time. Diagnosis applications are one domain that initially lacks complete knowledge, but then the knowledge about the domain can be increased over time, which leads to better diagnoses.

Learning is not part of rule-based reasoning systems, so new rules cannot be learned automatically and must usually be manually added to the system. Also a significant amount of training data is needed to extract the rules that are not obvious in a domain. Case-based reasoning systems, however, typically require only a few examples to help to solve the problem. Although the solution may not be completely accurate, the feedback in these systems can help to improve the accuracy of solutions over time.

Rule-based systems cannot provide an answer for a new problem entering the system that does not match any of the rule sets in the system. Case-based reasoning can solve new problems with the help of previous problem-solutions that have some part in common with the new one, and by combining solutions in some logical way.

The above problems with diagnosis applications also exist with planning applications. It is hard in systems in the planning domain to gather all the knowledge about actions and conditions. Finding a plan from scratch is very time consuming especially when the plan space is large. Demand for new plans which have not been made before with the system is not unusual. All these features of planning domain applications, make using case-based reasoning a good option for solving planning problems.

8. Summary

Case based reasoning was reviewed in this paper. We have described the basic parts in case base reasoning: case representation, case base models, case retrieval, reuse, revision and retention. For each part we overviewed of the research in the area and tried to compare the different methods by including the advantages and disadvantages of each method that has been used. This advantages and disadvantages can help for the CBR system developers for deciding on the method they can use in their system. Also they can be used for creating hybrid methods for different parts according to the requirements of the systems. For existing memory models and retrieval methods, we constructed taxonomy of models according to the literature.

In the last part of the paper we had a look on two groups of applications: diagnosis and planning. These two groups of applications were selected because of the project we have in our lab that includes both diagnosis and planning on solving the diagnosed problem, and we want to use case based reasoning for it. In each group we overviewed the methods that have been used and the reasons that they are most common in this group of applications. Open problems in CBR have been reviewed in the last part.

This survey helped us to find the available methods according to our project's requirements, how to combine different methods for different parts of the system and how to improve the performance of different parts by changing the method.

References

[1] B. Goertzel and C. Pennachin, Eds., *Artificial General Intelligence*. Springer, 2005.

[2] I. Watson and F. Marir, "Case-Based Reasoning: A Review," *Knowledge Engineering Review*, vol. 9, no. 4, 1994.

[3] C. W. Lehman and M. J. Willshire, "A Rule-Based Expert System for Diagnosis of Convergence Problems in Circuit Simulation," in *18th International Conference on Software Engineering and Knowledge Engineering*, San Fransisco, 2006, pp. 57-60.

[4] M. S. Kandil, S. M. Debeiky, and N. E. Hasanein, "A Hybrid Mathematical and Rule-Based System for Transmission Network Planning in a Deregulated Environment," in *Power Engineering Society Meeting*, Vancouver, 2001, pp. 1451-1456.

[5] Z. Ismail, A. Yahya, and K. A. Mahpol, "Forecasting Peak Load Electricity Demand Using Statistical and Rule-Based Approach," *American Journal of Applied Science*, vol. 6, no. 8, pp. 1618-1625, 2009.

[6] M. S. Kandil, S. M. Debeiky, and N. E. Hasanein, "Rule-Based System for Determining Unit Locations of Developed Generation Expansion Plan for Transmission Planning," *Generation, Transmission and Distribution*, vol. 147, no. 1, pp. 62-68, 2000.

[7] S. Slade, "Case-based Reasoning: A Research Paradigm," *AI Magazine*, vol. 12, pp. 42-55, Jan. 1991.

[8] I. Watson, "An Introduction to Case-Based Reasoning," *Progress in Case-Based Reasoning, Lecture Notes in Computer Science*, vol. 1020, pp. 1-16, 1995.

[9] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variation, and System Approaches," *AI communication*, vol. 7, no. 1, pp. 39-59, 1994.

[10] J. L. Kolonder and D. Leake, "A Toutorial Introduction to Case-Based Reasoning," in *Case-Based Reasoning: Experiences,Lessons, and Future Directions*, D. B. Leake, Ed. California, US: MIT press, 1996, ch. 2, pp. 31-65.

[11] C. Tsatsoulis and A. B. Williams, "Case-Based Reasoning," *Knowledge-Based Systems*, vol. 3, pp. 807-837, 2000.

[12] X. Qin and W. C. Regli, "A Study in Applying Case-Based Reasoning to Engineering Design," *Artificial Intelligence*, vol. 17, no. 3, pp. 235-252, 2003.

[13] S. Craw, N. Wiratunga, and R. C. Rowe, "Learning Adaptation Knowledge to Improve Case-Based Reasoning," *Artificial Intelligence*, vol. 170, pp. 1175-1192, 2006.

[14] Q. Su, "Applying Case-Based Reasoning in Assembly Sequence Planning," *International Journal of Production Research*, vol. 45, no. 1, pp. 29-47, Jan. 2007.

[15] S. Otim, "A Case-Based Knowledge Management System for Disaster Management: Fundamental Concepts," in *3rd International Information Systems for Crisis Response and Management (ISCRAM) conference*, Newark, 2006, pp. 598-604.

[16] C. Marling, M. Tomko, M. Gillen, D. Alexander, and D. Chelberg, "Case-Based Reasoning for Planning and World Modeling in the RoboCup Small Size League," in *IJCAI-03 Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments: World Modeling, Planning, Learning, and Communicating*, Mexico, 2003.

[17] X. Song, S. Petrovic, and S. Sundar, "A Case-Based Reasoning Approach to Dose Planning in Radiotherapy," in *7th International Conference on Case-Based Reasoning, ICCBR '07*, Belfast, 2007, pp. 348-357.

[18] C. Tsatsoulis and P. L. Kashyap, "Case-Based Reasoning and Learning in Manufacturing with TOLTEC Planner," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 4, pp. 1010-1023, Jul. 1993.

[19] I. Bichindaritz, E. Kansu, and K. M. Sullivan, "Case-Based Reasoning in CARE-PARTNER: Gathering Evidence for Evidence-Based Medical Practice," in *4th European Workshop on Case Based Reasoning*, Dublin, 1998, pp. 334-345.

[20] B. Lopez and E. Plaza, "Case-Based Learning of Plans and Goal States in Medical Diagnosis," *Artificial Intelligence in Medicine*, vol. 9, pp. 29-60, 1997.

[21] D. B. Leake, A. Kinley, and D. Wilson, "Learning to Improve Case Adaption by Introspective Reasoning and CBR," in *1st International Conference on Case-Based Reasoning and Development*, Sesimbra, 1995, pp. 229-240.

[22] S. Ontanon, K. Mishra, N. Sugandh, and A. Ram, "Case-Based Planning and Execution for Real-Time Strategy Games," in *7th International Conference on Case-Based Reasoning and Development*, Belfast, 2007, pp. 164-178.

[23] H. Zhuo, Q. Yang, and L. Li, "Constraint-Based Case-Based Planning Using Weighted MAX-SAT," in *8th International Conference on Case-Based Reasoning*, vol. 5650, Seattle, 2009, pp. 374-388.

[24] C.-H. L. Lee, K. Y.-R. Cheng, and A. Liu, "A Case-Based Planning Approach for Agent-Based Service-Oriented Systems," in *Systems, Man and Cybernetics*, Singapour, 2008, pp. 625-630.

[25] J. F. De Paz, S. Rodriguez, J. M. Corchado, and J. Bajo, "Dynamic Case-Based Planning," in *8th International Conference on Computational and Mathematical Methods in Science and Engineering*, Murcia, Spain, 2008.

[26] C.-H. L. Lee and A. Liu, "A Case-Based Planning Approach for Mobile Agents Migration Guidance," in *Systems Man and Cybernetics*, Istanbul, 2010, pp. 4002-4007.

[27] S. Ontanon, K. Mishra, N. Sugandh, and A. Ram, "On-Line Case-Based Planning," *Computational Intelligence*, vol. 26, no. 1, pp. 84-119, 2010.

[28] A. A. Niknafs, M. E. Shiri, and M. M. Javidi, "A Case-Based Reasoning Approach in E-Tourism: Tour Itinerary Planning," in *14th International Workshop on Databaseand Expert Systems*

Application (DEXA '03), Prague, 2003, pp. 818-822.

[29] M. D'Aquin, J. Lieber, and A. Napoli, "Adaptation Knowledge Aquisition: A Case Study for Case-Based Decision Support in Oncology," *Computational Intelligence*, vol. 22, no. 3/4, pp. 161-176, 2006.

[30] K. J. Hammond, "Case-Based Planning: A Framework for Planning from Experience," *Cognitive Science*, vol. 14, pp. 385-443, 1990.

[31] C. Mechiors and L. M. R. Tarouco, "Fault Management in Computer Networks using Case-Based Reasoning : DUMBO System," in *3rd International Conference on Case-Based Reasoning and Development*, vol. 1650, Seeon Monastery, 1999, pp. 510-524.

[32] A.-B. M. Salem, "Case-Based Reasoning Technology for Medical Diagnosis," *World Academy of Science, Engineering and Technology*, vol. 31, pp. 9-13, 2007.

[33] J. Hunt, "Case-Based Diagnosis and Repair of Software Faults," *Expert Systems*, vol. 14, no. 1, pp. 15-23, Feb. 1997.

[34] F. DIaz, F. Fdez-Riverola, and L. M. Corchado, "GENE-CBR: A Case-Based Reasoning Tool for Cancer Diagnosis Using Microarray Data Sets," *Computational Intelligence*, vol. 22, no. 3/4, pp. 254-268, 2006.

[35] S. G. Lee and Y. C. Ng, "Hybrid Case-Based Reasoning for On-Line Product Fault Diagnosis," *International Journal of Manifacturing Technology*, vol. 27, pp. 833-840, 2006.

[36] S. Montani and C. Anglano, "Case-Based Reasoning for Autonomous Service Failure Diagnosis and Remediation in Software Systems," in *European Conference on Case Based Reasoning*, vol. 4106, Oludeniz, 2006, pp. 489-503.

[37] A.-B. M. Salem, M. Roushdy, and R. A. HodHod, "A Case-Based Expert System for Supporting Diagnosis of Heart Diseases," *Artificial Intelligence and Machine Learning Journal*, vol. 5, no. 1, pp. 33-35, 2005.

[38] E. Olsson, P. Funk, and N. Xiong, "Fault Diagnosis in Industry using Sensor Readings and Case-Based Reasoning," *Journal of Intelligent and Fuzzy Systems*, vol. 15, no. 1, pp. 41-46, 2004.

[39] E. Golobardes, X. Llora, M. Salamo, and J. Marti, "Computer Aided Diagnosis with Case-Based Reasoning and Genetic Algorithms," *Knowledge-Based Systems*, vol. 15, pp. 45-52, 2002.

[40] C. Maurente, E. O. Edye, S. H. Delgado, and D. R. Garcia, "Evaluation of Case-Based Reasoning for Clinical Decision Support Systems Applied to Acute Meningitis Diagnose," in *Innovations and Advances in Computer Sciences and Engineering*, T. Sobh, Ed. Newyork, USA: Springer, 2010, pp. 259-264.

[41] W. Cheetam, A. Varma, and K. Goebel, "Case-Based Reasoning at General Electric," in *14th International Florida Artificial Intelligence Research Society Conference*, Key west, 2001, pp. 93-97.

[42] S. L. Wang and S. H. Hsu, "A Web-Based CBR Knowledge Management System for PC Troubleshooting," *International Journal of Manifacturing Technology*, vol. 23, pp. 532-540, 2004.

[43] J. M. G. Guiu, E. G. Ribe, E. B. Mansilla, and X. L. Fabrega, "Automatic Diagnosis with Genetic Algorithms and Case-Based Reasoning," *Artificial Intelligence*, vol. 13, pp. 367-372, 1999.

[44] M. Baig, "Case-Based Reasoning - An Effective Paradigm for Providing Diagnostic Support for Stroke Patient," Msc. thesis, Queen's university, Kingston, 2008.

[45] W. Wang and T. Yu, "Remote Fault Diagnosis System of Ultrahigh Speed Grinding Based on Multi-Agent," *Advanced Materials Research*, vol. 76-78, pp. 67-71, 2009.

[46] Y.-T. Tsai, "Applying a Case-Based Reasoning Method for Fault Diagnosis During Maintenance," *Journal of Mechanical Engineering Science*, vol. 223, no. 10, pp. 2431-2441, 2009.

[47] L. Derere, "Case-Based Reasoning : Diagnosis of Faults in Complex Systems through Reuse of

Experience," in *International Test Conference 2000*, Atlantic city, 2000, pp. 27-34.

[48] P. W. Grant, P. M. Harris, and L. G. Moseley, "Fault Diagnosis for Industrial Printers using Case-Based Reasoning," *Engineering Application of Artificial Intelligence*, vol. 9, no. 2, pp. 163-173, 1996.

[49] P. Koton, "Reasoning about Evidence in Causal Explanation," in *7th National Conference on Artificial Intelligence*, Saint Paul, 1988, pp. 256-261.

[50] M. Devaney, A. Ram, H. Qui, and J. Lee, "Preventing Failures by Mining Maintenance Logs with Case-Based Reasoning," in *59th Meeting of the Society for Machinery Failure Prevention Technology*, Virginia beach, 2005.

[51] J. N. K. Liu and T. T. S. Leung, "A Web-Based CBR Agent for Financial Forecasting a Workshop Program," in *4th International Conference on Case-Based Reasoning*, Vancouver, 2001, pp. 243-253.

[52] S. Schmitt, "simVar: A Similarity-Influenced Question Selection Criterion for e-Sales Dialogs," *Artificial Intelligence Review*, vol. 18, no. 3-4, pp. 195-221, 2002.

[53] H. Shimazu, "ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops," *Artificial Intelligence Review*, vol. 18, no. 3-4, pp. 223-244, 2002.

[54] B. Limthanmaphon and Y. Zhang, "Web Service Composition with Case-Based Reasoning," in *14th Australasian Conference*, Adelaide, 2003, pp. 201-208.

[55] S. Lajmi, C. Ghadira, K. Ghedira, and D. Benslimane, "WeSCo_CBR: How to Compose Web Services via Case-Based Reasoning," in *IEEE International Conference on E-Business Engineering (ICEBE '06)*, Shanghai, 2006, pp. 618-622.

[56] R. Bergmann, J. Kolonder, and E. Plaza, "Representation in Case-Based Reasoning," *The Knowledge Engineering Review*, vol. 20, no. 3, pp. 209-213, 2006.

[57] L. Coyle, C. Hayes, and P. Cunningham, "Representing Cases for CBR in XML," Trinity College Dublin, Department of Computer Science, Dublin, Technical Report, 2002.

[58] I. Bichindaritz, "Memory Structures and Organization in Case-Based Reasoning," in *Case-Based Reasoning on Images and Signals*, P. Perner, Ed. Berlin: Springer, 2008, ch. 6, pp. 175-194.

[59] M. Malek, "A Connectionist Indexing Approach for CBR Systems," in *1st International Conference on Case-Based Reasoning Research and Development*, Sesimbra, 1995, pp. 520-527.

[60] Z. Zhang and Q. Yang, "Towards Lifetime Maintenance of Case Base Indexes for Continual Case Based Reasoning," in *International Conference on AI Methodologies Systems and Applications*, Sozopol, 1998, pp. 489-500.

[61] S. Wess, K.-D. Althoff, and G. Derwand, "Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning," in *1st European Workshop, Topics in Case-Based Reasoning*, Kaiserslautern, 1993, pp. 67-81.

[62] B. Diaz-Agudo and P. A. Gonzales-Calero, "Classification Based Retrieval Using Formal Concept Analysis," in *4th International Conference on Case-Based Reasoning, Research and Developement*, Vancouver, 2001, pp. 173-188.

[63] R. Bergmann and A. Stahl, "Similarity Measures for Object-Oriented Case Represenatation," in *4th European Workshop on Advances in Case-Based Reasoning*, Duplin, 1998, pp. 25-36.

[64] B. Smyth and E. McKenna, "Footprint-Based Retrieval," in *3rd International Conference on Case-Based Reasoning*, Munich, 1999, pp. 343-357.

[65] M. Lenz and H.-D. Burkhard, "Case Retrieval Nets: Basic Ideas and Extensions," in *KI-96: Advances in Artificial Intelligence, 20th Annual German Conference on Artificial Intelligence*, Dresden, 1996, pp. 227-239.

[66] J. W. Schaaf, "Fish and Shrink. A Next Step Towards Efficient Case Retrieval in Large Scaled Case Bases," in *3rd European Workshop on Advances in Case-Based Reasoning*, Lausanne, 1996, pp. 362-376.

[67] H. Guo, X. Zhou, and Y. Zhu, "Knowledge Aquisition Based on Rough Set and Data Mining," in *International Conference on Future BioMedical Information Engineering*, Sanya, 2009, pp. 126-128.

[68] C.-C. Hsu and Y.-H. Huang, "Case Mining from Raw Data for Library Construction," in *9th Joint Conference on Information Science*, Kaohsiung, 2006, pp. 1091-1094.

[69] B. Smyth and M. T. Keane, "Retrieving Adaptable Cases: The Role of Adaptation Knowledge in Case Retrieval," in *1st European Workshop on Topics in Case-Based Reasoning*, Kaiserslautern, 1993, pp. 209-220.

[70] S. Craw, "Introspective Learning to Build Case-Based Reasoning (CBR) Knowledge Containers," in *3rd International Conference on Machine Learning and Datamining for Pattern Recognition*, Leipzig, 2003, pp. 1-6.

[71] N. Arshadi and I. Jurisica, "Data Mining for Case-Based Reasoning in High-Dimentional Biological Domains," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 8, pp. 1127-1137, Aug. 2005.

[72] R. L. De Mantaras, et al., "Retrieval,Reuse, Revision and Retention in Case-Based Reasoning," *The Knowledge Engineering Review*, vol. 20, no. 3, pp. 215-240, 2006.

[73] E. Simoudis and J. Miller, "Validated Retrieval in Case-Based Reasoning," in *8th National Conference on Artificial Intelligence*, Boston, 1990, pp. 310-315.

[74] D. W. Patterson, N. Rooney, and M. Galushka, "Efficient Similarity Determination and Case Construction Techniques for Case-Based Reasoning," in *6th European Conference on Advances in Case-Based Reasoning (ECCBR 02)*, Aberdeen, 2002, pp. 292-305.

[75] M. Galushka and D. Patterson, "Intelligent Index Selection for Case-Based Reasoning," *Knowledge-Based Systems*, vol. 19, pp. 625-638, 2006.

[76] N. Stephane, R. Hector, and L. L. J. Marc, "Effective Retrieval and New Indexing Method for Case-Based Reasoning: Application in Chemical Design," *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 880-894, 2010.

[77] K. Borner, "Structural Similarity as Guidance in Case-Based Design," in *1st European Workshon on Topics in Case-Based Reasoning*, Kaiserslautern, 1993, pp. 197-208.

[78] A. Aamodt, "Knowledge-Intensive Case-Based Reasoning in CREEK," in *7th European Conference on Advances in Case-Based Reasoning*, Madris, 2004, pp. 793-850.

[79] B. Mogouie and R. Bergmann, "Similarity Assessment for Generalizied Cases by Optimization Methods," in *6th European Conference on Advances in Case-Based Reasoning*, Aberdeen, 2002, pp. 249-263.

[80] A. Tartakovski, M. Schaaf, R. Maximini, and R. Bergmann, "MINLP Based Retrieval of Generalized Cases," in *7th European Conference on Advances in Case-Based Reasoning*, Madrid, 2004, pp. 404-418.

[81] A. Tartakovski, M. Schaaf, and R. Maximini, "Applying Generalized Cases to Retrieval and Configuration of Life Insurance Policies," in *Wissensmanagement 2005 (LNAI 3782)*, Kaiserslautern, 2005, pp. 293-303.

[82] S. Petrovic, G. Kendall, and Y. Yang, "A Tabu Search Approach for Gragh-Structured Case Retrieval," in *Starting Artificial Intelligence Researchs Syposium*, Lyon, 2002, pp. 55-64.

[83] J. L. Arcos and R. L. de Mantaras, "Perspectives: A Declarative Bias Mechanism for Case Retrieval," in *2nd International Conference on Case-Based Reasoning and Developement*, Rhode Island, 1997, pp. 279-290.

[84] A. A. Assali, D. Lenne, and B. Debray, "Case Retrieval in Ontology-Based CBR Systems," in

32nd Annual Conference on Artificial Intelligence (Ki 2009), Paderborn, 2009, pp. 564-571.

[85] B. Smyth and M. T. Kean, "Adaptation -Guided Retrieval: Questioning the Similarity Assumption in Reasoning," *Artificial Intelligence*, vol. 102, pp. 249-293, 1998.

[86] B. Mougouie, M. M. Richter, and R. Bergmann, "Diversity-Conscious Retrieval from Generalized Cases: A Branch and Bound Algorithm," in *5th International Conference on Case-Based Reasoning and Developments*, Trondheim, 2003, pp. 319-331.

[87] B. Smyth and P. McClave, "Similarity vs. Diversity," in *4th International Conference on Case-Based Reasoning and Developement*, Vancouver, 2001, pp. 347-361.

[88] D. Bridge and A. Ferguson, "Diverse Product Recommendations Using an Expressive Language for Case Retrieval," in *6th European Conference on Advances in Case-Based Reasoning*, Aberdeen, 2002, pp. 43-57.

[89] D. McSherry, "Diversity-Conscious Retrieval," in *6th European Conference on Advances in Case-Based Reasoning*, Aberdeen, 2002, pp. 219-233.

[90] D. McSherry, "Similarity and Compromise," in *5th International Conference on Case-Based Reasoning and Developement*, Trondheim, 2003, pp. 291-305.

[91] D. McSherry, "Balancing User Satisfaction and Cognitive Load in Coverage-Optimised Retrieval," *Knowledge-Based Systems*, vol. 17, pp. 113-119, 2004.

[92] K. J. Hammond, "Explaining and Repairing Plans that Fail," *Artificial Intelligence*, vol. 45, no. 1-2, pp. 173-228, Sep. 1990.

[93] R. Barletta and W. Mark, "Explanation-Based Indexing of Cases," in *7th National Conference on Artificial Intelligence (AAAI-88)*, Saint Paul, 1988, pp. 541-546.

[94] D. Doyle, P. Cunningham, D. Bridge, and Y. Rahman, "Explanation Oriented Retrieval," in *7th European Conference on Advances in Case-Based Reasoning*, Madrid, 2004, pp. 157-168.

[95] B. Fuchs, J. Lieber, A. Mille, and A. Napoli, "An algorithm for adaptation in case-based reasoning," in *European Conference on Artificial Intelligence*, Lisbon, 2000, pp. 45-49.

[96] K. Hanney and M. T. Keane, "Learning Adaptation Rules from a Case Base," in *3rd European Conference on Advances in Case-Based Reasoning*, Lausanne, 1996, pp. 179-192.

[97] H. Li, et al., "Adaptation Rule Learning for Case-Based Reasoning," *Concurrancy and Computation: Pratice and Experience*, vol. 21, no. 5, pp. 673-689, Apr. 2009.

[98] M. D'Aquin, et al., "Case Base Mining for Adaptation Knowledge Aquisition," in *International Joint Conference on Artificial Intelligence,IJCAI-07*, Hyderabad, 2007, pp. 750-755.

[99] L. Spalazzi, "A Survey on Case-Based Planning," *Artificial Intelligence Review*, vol. 16, pp. 3-36, 2001.

[100] E. C. C. Tsang, S. C. K. Shiu, X. Z. Wang, and K. Ho, "Case Base Reduction using Learned Local Feature Weights," in *he Joint 9th IFSA World Congress and 20th NAFIPS International Conference: Fuzziness and Soft Computing in the New Millenium*, Vancouver, 2001, pp. 2965-2970.

[101] M. Salamo and E. Golobardes, "Hybrid Deletion Policies for Case Base Maintenance," in *16th International FLAIRS Conference*, St. Augustine, 2003, pp. 150-154.

[102] B. Smyth and M. T. Keane, "Remembering to Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems," in *14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, Montreal, 1995, pp. 377-383.

[103] B. Smyth and E. McKenna, "Building Compact Competent Case Bases," in *3rd International Conference on Case-Based Reasoning and Developement (ICCBR '99)*, Seeon Monastery, 1999, pp. 329-342.

[104] N. H. Lu, G.-q. Zhang, and J. Lu, "Detecting Change via Competence Model," in *18th International Conference on Case-Based Reasoning, Research and Developement (ICCBR 2010)*,

Alessandria, 2010, pp. 201-212.

[105] M. K. Haouchine, B. Chebel-Morello, and N. Zerhouni, "Competence-Preserving Case-Deletion Strategy for Case Base Maintenance," in *9th European Workshop on Advances in Case-Based Reasoning 2008 (ECCBR'08)*, Trier, 2008, pp. 171-183.

[106] D. B. Leake and D. C. Wilson, "Remembering Why to Remember: Performance-Guided Case Base Maintenance," in *5th European Workshop on Advances in Case-Based Reasoning*, Trento, 2000, pp. 161-172.

[107] J. Zhu and Q. Yang, "Remembering to Add: Competence-Preserving Case-Addition Policies for Casebase Maintenance," in *International Joint Conference in Artificial Intelligence IJCAI99*, Stockholm, 1999, pp. 234-241.

[108] A. Lawanna and J. Daengdej, "Methods for Case Maintenance in Case-Based Reasoning," *International Journal of Computer and Information Engineering*, vol. 4, no. 1, pp. 10-18, 2010.

[109] S. Ontanon and E. Plaza, "Justification-Based Selection of Training Examples for Case Base Reduction," in *15th European Conference on Machine Learning (ECML 2004)*, Pisa, 2004, pp. 310-321.

[110] Z.-W. Ni, Y. Liu, F.-G. Li, and S.-L. Yang, "Case Base Maintenance Based on Outlier Data Mining," in *4th International Conference on Machine Learning and Cybernetics*, Guanzhou, 2005, pp. 2861-2864.

[111] M. Salamo and M. Lopez-Sanchez, "Adaptive Case-Based Reasoning Using Retention and Forgetting Strategies," *Knowledge-Based Systems*, vol. 24, pp. 230-247, 2011.

[112] H. Romdhane and L. Lamontagne, "Forgetting Reinforced Cases," in *9th European Conference on Advances in Case-Based Reasoning*, Trier, 2008, pp. 474-486.

[113] M. Salamo and E. Golobardes, "Deleting and Building Sort Out Techniques for Case Base Maintenance," in *6th European Conference on Case Based Reasoning (ECCBR 2002)*, Aberdeen, 2002, pp. 365-379.

[114] S. C. K. Shiu, E. C. C. Tsang, and D. S. Yeung, "Maintaining Case-Based Expert Systems using Fuzzy Neural Network," in *1999 IEEE Int. Conf. on Systems, Man and Cybernetics*, Tokyo, 1999, pp. 424-428.

[115] Q. Yang and J. Wu, "Keep it Simple: A Case Base Maintenance Policy Based on Clustering and Information Theory," in *13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence on advances in Artificial Intelligence*, Montreal, 2000, pp. 102-114.

[116] S. Massie, S. Craw, and N. Wiratunga, "When Similar Problems don't have Similar Solutions," in *7th International Conference on Case-Based Reasoning*, Belfast, 2007, pp. 92-106.

[117] K. Racine and Q. Yang, "On the Consistency Management of Large Case Bases: the Case for Validation," in *AAAI '96 -Verification and Validation Workshop*, Portland, 1996.

[118] L. Portinale, P. Torasso, and P. Tavano, "Dynamic Case Memory Management," in *13th European Conference on Artificial Intelligence (ECAI 98)*, Brighton, 1998, pp. 73-78.

[119] L. Portinale, P. Torasso, and P. Tavano, "Speed-Up, Quality and Competence in Multi-Model Case-Based Reasoning," in *3rd International Conference and Case-Based Reasoning and Developements (ICCBR '99)*, Seeon Monastery, 1999, pp. 303-317.

[120] D. W. Patterson, M. Galushka, and N. Rooney, "Charachterisation of a Novel Indexing Technique for Case-Based Reasoning," *Artificial Intelligence Review*, vol. 23, pp. 359-393, 2005.

[121] T. M. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[122] E. Simoudis and P. Miller, "Automated Support for Developing Retrieve-and-Propose System," in *11th International Conference on Artificial Intelligence*, Washington, 1993, pp. 147-152.

[123] D. Wang, Y. Xiang, G. Zou, and B. Zhang, "Research on Ontology-Based Case Indexing in CBR," in *International Conference on Artificial Intelligence and Computational Intelligence*, Las Vegas, 2009, pp. 238-241.

[124] D. W. Patterson, N. Rooney, M. Galushka, and S. Anand, "Towards Dynamic Maintenance of Retrieval Knowledge in CBR," in *15th Internation FLAIRS Conference*, Pensacola Beach, 2002, pp. 126-131.

[125] K.-S. Kim and I. Han, "The Cluster-Indexing Method for Case-Based Reasoning Using Self-Organizing Maps and Learning Vector Quantization for Bond Rating Cases," *Expert Systems with Applications*, vol. 21, pp. 147-156, 2001.

[126] D. Y. Sha and C. .-H. Liu, "Development and Evaluation of a Tree-Indexing Approach to Improve Case-Based Reasoning : Illustrated using the Due Date Assignment Problem," *International Journal of Production Reseach*, vol. 44, no. 15, pp. 3033-3049, Aug. 2006.

[127] W.-C. Chen, S.-S. Tseng, L.-P. Chang, T.-P. Hong, and M.-F. Jiang, "A Parallelized Indexing Method for Large-Scale Case-Based Reasoning," *Expert Syetems with Application*, vol. 23, pp. 95-102, 2002.

[128] J. Daengdej, D. Lukos, E. Tsui, P. Beinat, and L. Prophet, "Dynamically Creating Indeces for Two Million Cases: A Real World Problem," in *3rd European Workshop on Advanced in Case-Based Reasoning (EWCBR`96)*, Lausanne, 1996, pp. 105-119.

[129] S. Fox and D. B. Leake, "Learning to Refine Indexing by Introspective Reasoning," in *First International Conference on Case-Based Reasoning and Developement (ICCBR 1995)*, Sesimbra, 1995, pp. 430-440.

[130] (2011, Apr.) Wikipedia, Diagnosis Definition. [Online]. http://en.wikipedia.org/wiki/Diagnosis

[131] M. H. Goker, R. J. Howlett, and J. E. Price, "Case-Based Reasoning for Dianosis Applications," *The Knowledge Engineering Review*, vol. 20, no. 3, pp. 277-281, 2006.

[132] A. Carneiro, et al., "DBSitter: An Intelligent Tool for Database Administration," in *15th International Conference on Database and Expert Systems Applications (DEXA 2004)*, Zaragoza, 2004, pp. 171-180.

[133] M. T. Cox, H. Munoz-Avila, and R. Bergmann, "Case-Based Planning," *The Knowledge Engineering Review*, vol. 20, no. 3, pp. 283-287, 2006.

[134] R. Bergmann, H. Munoz-Avila, M. Veloso, and E. Melis, "Case-based Reasoning Applied for Planning," in *Case-Based Reasoning Technology, From Foundations to Applications*, 1998, pp. 169-200.

[135] P. Liberatore, "On the Complexity of Case-Based Planning," *Journal of Experimantal and Technical Artificial Intelligence*, vol. 17, no. 3, pp. 283-295, Sep. 2005.

[136] H. Munoz-Avila and M. T. Cox, "Case-Based Plan Adaptation: An Analysis and Review," *IEEE Expert / IEEE Intelligent Systems*, vol. 23, no. 4, pp. 75-81, 2008.

[137] M. U. Ahmed, S. Begum, N. Xiong, and P. Funk, "Case-Based Reasoning for Medical and Industrial Decision Supprt Systems," *Successful Case-Based Reasoning Applications*, vol. 1, pp. 7-52, 2010.