A Tutorial on Case-Based Reasoning

Julie Main, Tharam Dillon and Simon Shiu

Abstract

This tutorial article introduces the concepts and applications of Case-based Reasoning (CBR) systems. The first section briefly describes what is CBR, when to use CBR and why use CBR. The second section looks at the description and indexing of cases in CBR systems. The retrieval and adaptation processes for finding solutions are outlined in section three. Learning and maintenance of CBR, due to the changes in domain knowledge and task environments over time, are given in section four. The role of soft computing in CBR is briefly described in section five. The final section gives some examples of successful CBR applications in different areas.

1 What is Case-Based Reasoning?

1.1 Introduction

A short definition of case-based reasoning is that it is a methodology for solving problems by utilizing previous experiences. It involves retaining a memory of previous problems and their solutions and, by referencing these, solve new problems. Generally[1], a case-based reasoner will be presented with a problem. It may be presented by either a user or another program or system. The case-based reasoner then searches its memory of past cases (the case base) and attempts to find a case that has the same problem specification as the current case. If the reasoner cannot find an identical case in its case base, it will attempt to find the case or cases in the case base that most closely match the current query case.

In the situation where a previous identical case is retrieved, presuming its solution was successful, it can be returned as the current problem's solution. In the more likely case that the retrieved case is not identical to the current case, an adaptation phase occurs. In adaptation, the differences between the current case and the retrieved case must first be identified and then the solution associated with the retrieved case modified taking into account these differences. The solution returned in response to the current problem specification may then be tried in the appropriate domain setting.

The structure of a case-based reasoning system therefore is usually devised in a manner that reflects these separate stages. At the highest level

---

[1] This tutorial describes an average case-based reasoning system. There are many variations have been used in implementations, which we do not discuss.

Soft Computing in Case Based Reasoning, (Eds.) Sankar K Pal, Tharam Dillon and Daniel Yeung, Springer-Verlag (London) Ltd, 2001, pp.1-28.

a case-based reasoning (CBR) system can be thought of as a black box (see Figure 1) that incorporates the reasoning mechanism and the external facets:

- the input specification, (or problem case)
- the output suggested solution
- the memory of past cases that are referenced by the reasoning mechanism.

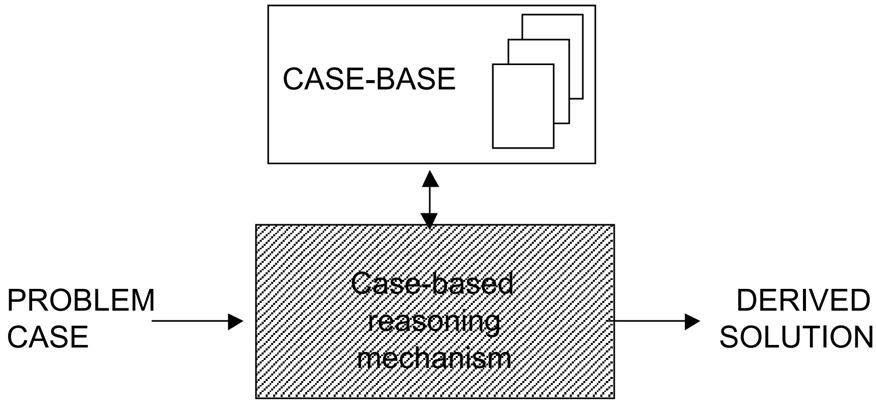

Figure 1. A CBR system

In most CBR systems, the case-based reasoning mechanism, alternatively referred to as the problem solver or reasoner, has an internal structure divided into two major parts; the case retriever and the case reasoner as shown in Figure 2.

The case retriever's task is to find the appropriate cases in the case base while the case reasoner uses the retrieved cases to find a solution to the given problem description. This reasoning generally involves both determining the differences between the retrieved cases and the current query case; and modifying the retrieved solution appropriately, reflecting these differences. This reasoning part itself may or may not retrieve further cases or portions of cases from the case base.

Thus, we begin to see the internal structure of the CBR system. This approach in case based reasoning can be contrasted with that used in other knowledge based systems such as rule based systems or combined frame-rule based systems. In rule based systems, one has a rule base consisting of a set of production rules of the form IF A THEN B where A is a condition and B an action. If the condition A holds, then action B is carried out. 'A' can be a composite condition consisting say of a conjunction of premises A1, A2, …An. In addition, the rule based system has an inference engine, which compares the data it holds in working memory with the condition parts of rules to determine which rules fire. Combined frame-rule based systems also utilize frames in addition to rule to capture stereotypical knowledge. These frames consist of Slots, which can have default values, actual values or attached daemons which when triggered use a procedure or a rule set to determine the required values. These rule based and combined frame-rule based systems require one to acquire the symbolic knowledge represented in these rules or frames by knowledge acquisition using manual knowledge engineering or automated knowledge acquisition tools. Sometimes one utilizes a model of the problem, as a basis of reasoning to a situation, such models can be qualitative or quantitative. Such systems are referred to as model based systems.

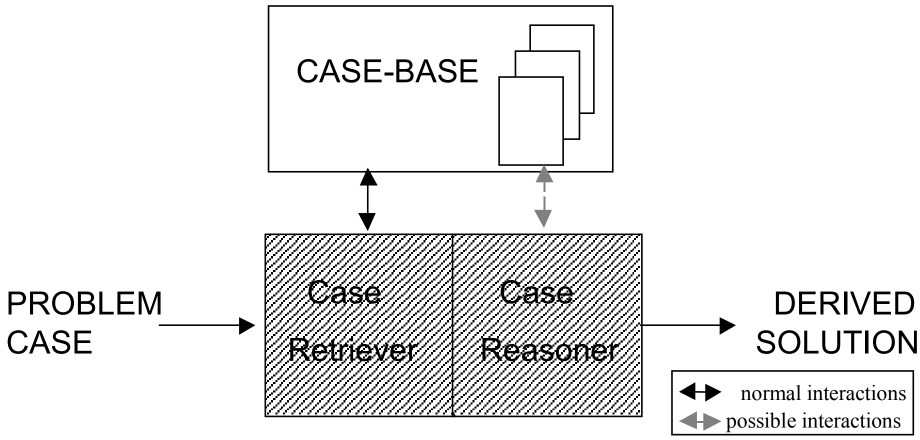

Figure 2. Two major components of a CBR system

Case-based reasoning systems are an alternative in many situations to rule-based systems. In many domains and processes, referring to cases as a means of reasoning can be an advantage due to the nature of this type of problem solving. One of the most time consuming aspects when developing a rule-based system is the knowledge acquisition task. Acquiring domain specific information and converting it into some formal representation can be a huge task and, in some situations, especially less understood domains the formalization of the knowledge cannot be done at all. Case-based systems usually require significantly less knowledge acquisition as it involves collecting a set of past experiences without the added necessity of extracting a formal domain model from these cases. In many domains, there are insufficient cases to extract a domain model. There is another benefit to CBR, which is that a system can be created with a small, or limited, amount of experience and incrementally developed, adding more cases to the case base as they become available.

The processes that make up case-based reasoning can be seen as a reflection of a particular type of human reasoning. In many situations, the problems humans encounter are solved with a human equivalent of CBR. When a person encounters a previously unexperienced situation or problem, they often refer to a past experience of a similar problem. This similar, previous experience may be one they have had or one another person has experienced. In the case that the experience was had by another human, the case will have been added to the (human) reasoner's memory via either an oral or written account of that experience.

In general, we have referred to case-based reasoning being applied to the solving of problems. Case-based reasoning can also be used in other ways, most notably that of arguing a point of view. For example, many students will come to their teacher or lecturer with various requests. These requests might be for an extension to a deadline or perhaps additional materials. It is a common experience of the teacher on refusal of one of these requests to have the student argue the point. One of the common techniques a student will use is to present evidence that in another course, or with another lecturer or teacher, their request was granted in a similar situation, with similar underlying rules.

This sort of reasoning is another way case-based reasoning systems may be implemented and is very common in law domains. Just as a barrister argues a point in court by reference to previous cases and the precedents they set, CBR systems can refer to a case base of court cases and find cases that have similar characteristics to the current one. The similarities may be in the whole case or only on certain points that led to a portion of the ruling.

Cases can be discovered therefore that may both support some portions of the current case but oppose other parts. Case-based systems that perform this sort of argument are generally referred to as interpretive reasoners.

1.2 What is a case

A case can be said to be the record of a previous experience or problem. The information recorded about this past experience will, by necessity, depend on the domain of the reasoner and the purpose to which the case will be put. In the instance of a problem solving CBR system, the details will usually include the specification of the problem and the relevant attributes of the environment that are the circumstances of the problem. The other vital part of the case is the solution that was applied in the previous situation. Depending on how the CBR system reasons with cases, this solution may include only the facts of the solution, or, additionally, the steps or processes involved in obtaining the solution. It is also important to include the achieved measure of success in the case description if the cases in the case base have achieved different degrees of success or failure.

When a comparison is made between the knowledge stored in a model/rule based system and that stored in a case base, it is apparent that the information in the case base is of a more specific nature than that of the model/rule based system. While the knowledge in a model/rule based system has been abstracted so that it is applicable in the widest variety of situations as possible, the knowledge contained in a case base remains specific to the case in which it is stored [1]. Because of the specific knowledge of a case base, we find that related knowledge and knowledge applicable in a specific circumstance is stored in close proximity. Thus, rather than drawing knowledge from a wide net, the knowledge needed to solve a specific problem case can be found grouped together in a few, or even one location.

The case base in the CBR system is the memory of all previous stored cases. There are three general areas that have to be considered when creating a case base.

- The structure and representation of the cases themselves
- The memory model used for organizing the entire case base
- The selection of indices which are used to identify each case

1.3 When to use case-based reasoning

While case-based reasoning is useful for many types of problems and in may different domains, there are times when it is not the most appropriate methodology to employ. There are a number of characteristics of problems and their domains that can be used to determine whether case-based reasoning is applicable [1][2][3]:

Does the domain have an underlying model?
If a process is random, or if the factors leading to the success or failure of a solution cannot be captured in the case description, any reasoning from past cases may be futile.

Are there exceptions and novel cases?
Domains without novel or exceptional cases may be better modeled with rules, which could be inductively determined from the cases.

Do cases recur?
If a case is not likely to be used in a subsequent problem, because of a lack of similarity then there is little, if any, value in storing the case. In these domains, when cases are not similar enough to be adapted then perhaps it would be better to build a model of the process of developing the solution, rather than a model of the solution domain.

Is there significant benefit in adapting past solutions?
One must consider whether there is a significant difference in the resources expended (time, processing, etc) between creating a solution to a problem from scratch and creating a solution through modifying a similar solution.

Are relevant previous cases obtainable?
It is possible to obtain the data that records the necessary characteristics of past cases? Do the recorded cases contain the features of the problem and

its context that influenced the outcome of the solution? Is the solution recorded in the detail necessary for it to be adapted in future?

If the answer to the majority of questions above is positive, then it is likely that a case-based reasoning may be applicable and relevant.

1.4 Why use CBR?

As many authors have discussed previously, when used in the appropriate situations, case-based reasoning offers many advantages. [2][3][4], In this section we summarize many of them. Some points have appeared in more detail in some of the above references, and often from varying points of view. The order in which they appear here is not indicative of their level of importance.

Reduction of the Knowledge Acquisition Task.
By eliminating the extraction of a model or a set of rules as is necessary in model/rule based systems, the knowledge acquisition tasks consists mainly of the collection of the relevant existing experiences/cases and their representation and storage.

Avoid repeating mistakes made in the past.
In systems that record failures as well as successes, and perhaps the reason for those failures, the system can use the information about what caused failures in the past to predict any failures in future. An example of such a system could be one which stores successful or failed lessons.

Graceful degradation of performance.
Some model based systems cannot even attempt to solve a problem on the boundaries of its knowledge or scope, or when there is missing or incomplete data. In contrast case-based systems can often have a reasonably successful attempt at solving these types of problem.

Able to reason in domains that have not been fully understood, defined or modeled.
While insufficient knowledge may exist about a domain to build a causal model of it or derive a set of heuristics for it, a case-based reasoner can function with only a set of cases from the domain. The underlying theory does not have to be quantified.

May be able to make predictions as to the probable success of a proffered solution.
Where information is stored regarding the level of success of past solutions, the reasoner may be able to predict the success of the suggested solution to a current problem. This may be done by referring both to the stored solutions and to the differences between the previous and current contexts of the solution.

Learn over time.

As CBR systems are used, they encounter more situations and create more solutions. If cases are tested in the real world and a level of success determined, these cases can be added into the case base to reason with in future. As we add cases, a CBR system should be able to reason in a wider variety of situations, and with a higher degree of refinement/success.

Reason in a domain with a small body of knowledge.

While a domain in which there is little known underlying knowledge and few cases from which to start limits the type of reasoning that can be done in it, a case based reasoner can start with the few known cases and incrementally increase its knowledge as cases are added to it. The addition of these cases will also cause the system to grow in the directions encountered by the system in its problem solving endeavors.

Reason with incomplete or imprecise data and concepts

As cases are retrieved not just when identical to the current query case but when they are within some measure of similarity, incompleteness and imprecision can be dealt with. While these factors may cause a slight degradation in performance due to the current and retrieved having increased disparity, reasoning can still continue

Avoid repeating all the steps that need to be taken to arrive at a solution.

In problem domains that require significant processes to carry out the creation of a solution from scratch, the modifying of a previous solution can significantly reduce this processing. By reusing a previous solution, the steps taken to reach the retrieved solution can be reused themselves.

Provide a means of explanation

Case-based reasoning can supply a previous case and its (successful) solution to convince a user, or justify to a user, a solution it is providing to their current problem. In most domains, there will be times when a user wishes to be reassured about the quality of the solution they are being given. By explaining how a previous case was successful in a situation, using the similarities between the cases and the reasoning involved in adaptation a CBR system can explain its solution to a user. Even in a hybrid system that may use multiple methods to find a solution, this explanation mechanism can augment the causal (or other) explanation given to the user.

Can be used in different ways.

The number of ways a CBR system can be implemented is almost unlimited. It can be used for many purposes as has been seen; for creating a plan, making a diagnosis, arguing a point of view, etc. As the data dealt with is likewise able to take many forms, so are the retrieval and adaptation methods. As long as stored past cases are being retrieved and adapted, case based reasoning is taking place.

Can be applied to a broad range of domains.
As will be discussed in the section on application areas, CBR has many areas of application. Due to the seemingly limitless number of ways of representing, indexing, retrieving and adapting cases, CBR can be applied to extremely diverse application domains.

Reflects human reasoning.
As there are many situations where we, as humans, use a form of case-based reasoning, it is not difficult to convince implementers, users and managers of the validity of the paradigm. Likewise, humans can understand a CBR system's reasoning and explanations and are able to be convinced of the validity of the solutions they are receiving. If the human user is wary of the validity of the received solution, they are less likely to use the solution given to them by the reasoner. The more critical the domain, the lower the chances of use, and the higher the level of the user's understanding and credulity will need to be.

2 Case Representation and Indexing

2.1 Case Representation

Cases in a case base can represent many different types of knowledge and store it in many different representational formats. The objective of a system will greatly influence what is stored. A case based reasoning system may be aimed at the creation of a new design or plan, the diagnosis of a new problem, or the argument of a point of view with precedents. In each type of system, a case may represent something different. The cases could be people, things or objects, situations, diagnoses, designs, plans or rulings among others. In many practical CBR applications, cases are usually represented as two unstructured sets of attribute value pairs, i.e. the problem and solution features [5]. However, the decision of what to represent can be one of the difficult decisions to make.

For example: In some sort of medical CBR system, that diagnosis a patient, a case could represent an individual's entire case history or be limited to a single visit to a doctor. In this situation the case may be a set of symptoms along with the diagnosis. It may also include a prognosis or treatment. If a case is a person then a more complete model is being used as this could incorporate the change of symptoms from one visit to the next. It is however harder to find and use cases in this format to search for a particular set of symptoms in a current problem and obtain a diagnosis/treatment. Alternatively if a case is simply a single visit to the doctor involving the symptoms at the time of that visit and the diagnosis of those symptoms, the changes in symptoms that might be a useful key in solving a problem may be missed.

In a situation such as the above, cases may need to be broken down and consist of sub-cases. For example, a case could be a person's medical history and could include all visits made by them to the doctor as sub cases. In an object-oriented representation this may be as follows (Figure 3) :

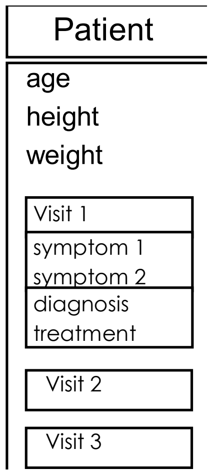

Figure 3. A patient case record

No matter what the case actually represents as a whole, the features of it have to be represented in some format. One of the advantages of case-based reasoning, is the flexibility it has in this regard. Depending on what types of features have to be represented, an appropriate implementation platform can be chosen. Ranging from simple Boolean, numeric and textual data to binary files, time dependent data, and relationships between data, CBR can be made to reason with all of them.

No matter what is stored, or the format it is represented in, a case must store that information that is relevant to the purpose of the system and which will ensure that the most appropriate case is retrieved in each new situation. Thus the cases have to include those features that will ensure that case will be retrieved in the most appropriate contexts.

In many CBR systems, all existing cases do not need to be stored. In these systems criteria are needed to decide which cases will be stored and which will be discarded. In the situation where two or more cases are very similar, only one case may need to be stored. Alternatively, it may be possible to create an artificial case that is a generalization of two or more actual incidents or problems. By creating generalized cases the most important aspects of a case need only be stored once.

When choosing a representation format for a case, there are many choices and many factors to consider. Some examples of representation formats that may be used include data base formats, frames, objects, and semantic networks. There are a number of factors that should to be considered when choosing a representation format for a case:

- The cases may have segments within them that form natural sub-cases or components. The forms this internal structure of a case may take, needs to be able to be represented in the chosen format.

- The content or features that describe a case have associated types and structures. These types have to be available or able to be created in the cases representation.

- The language or shell chosen in which to implement the CBR system. The choice of a shell may limit the formats that could be used for representation. It should also be noted that the choice of language or shell is going to be influenced by a number of factors. The availability of those shells or languages and the knowledge of the implementer of the possible choices are the primary influences here.

- The indexing and search mechanism planned. Cases have to be in a format which the case retrieval mechanism can deal with.

- The form in which cases are available or obtained. If the case base is to be formed from an existing collection of past experiences ease of being able to translate it into another appropriate form could be important.

Whatever format the cases are represented in, the collection of cases itself has to be structured in some way to facilitate the retrieval of the appropriate case when queried. Numerous approaches have been used for this. A flat case base is a common structure; in this method indices are chosen to represent the important aspects of the case and retrieval involves comparing the current cases features to each case in the case base. Another common case base structure is a hierarchical structure that stores the cases by grouping them to reduce the number of cases that have to be searched. The memory model. As with the form of case representation chosen will also depend on a number of factors.

- The representation used in the case base.

- The purpose to which the system is being put. For example, a hierarchical structure is a natural choice for a classification problem.

- The number and complexity of cases being stored. As the number of cases grows in the case base a structure such as a flat case base that is sequentially searched becomes more time consuming.

- The number of features that are used for matching cases during searches.

- Whether some cases are similar enough to group together. Where cases fall into groupings, some structuring facility may be useful.

- The amount of knowledge that is known about the domain will influence the ability to determine how similar cases are. If there is little domain knowledge then structuring cases is apt to be wrong.

2.2 Case Indexing

Case indexing refers to assigning indices to cases for future retrieval and comparisons. This choice of indices is important to being able to retrieve the right case at the right time. This is because the indices of a case will determine in which context it will be retrieved in future. These are some suggestions for choosing indices[1][6][7].

Indices must be both predictive and predictive in a useful manner. This means that they should reflect the important aspects of the case, the attributes that influenced the outcome of the case and also those which will describe the circumstances in which it is expected that they should be retrieved in the future.

Indices should be abstract enough to allow for that cases retrieval in all the circumstances in which the case will be useful, but not too abstract. When a case's indices are too abstract that case may be retrieved in too many situations, or too much processing would be required to match cases.

Although assigning indexes is still largely a manual process relies on human experts, various attempts of using automated methods were proposed in the literature. For example, Bonzano [8] uses inductive techniques for learning local weights of features by comparing similar cases in a case-base. Their method can determine which features are more important in predicting outcomes and improve retrieval. Bruninghaus [9] employs a factor hierarchy (a multi-level hierarchical knowledge that relates factors to normative concerns) in guiding machine learning programs to classify texts according to the factors and issues that apply. This method acts as an automatic filter of irrelevant information and structures the indexes into a factor hierarchy which represent the kinds of circumstances which are important to the users. Other methods include indexing cases by features and dimensions that are predictive across the entire problem domain [10]; by computing the differences between cases; adaptation guided indexing and retrieval [11] and explanation-based techniques.

3 Case Retrieval and Adaptation

3.1 Case Retrieval

Case retrieval is the process of finding within the case base those cases that are the closest to the current case. To carry out case retrieval there must be criteria that determine how a case is judged to be appropriate for retrieval and a mechanism to control how the case base is searched. The selection criteria is necessary to decide which case is the best one to retrieve, that is, to determine how close the current and stored cases are.

This criteria depends in part on what the case retriever is searching for. Most often the case retriever is searching for an entire case, the features of which will be compared to the current query case. There are however times when a portion of a case is required. This may be because no full case that exists and a solution is being built by selecting portions of multiple cases, or because a retrieved case is being modified by adopting a portion of another case in the case base.

The actual processes involved in retrieving a case from the case base depend very much on the memory model and indexing procedures used. Retrieval methods employed by researchers and implementers are extremely diverse, ranging from a simple nearest neighbor search to the use of intelligent agents. We discuss here the most common, traditional methods.

3.1.1 Nearest Neighbor Retrieval

In nearest neighbor retrieval, the case retrieved is chosen when the weighted sum of its features that match that query ease is greater than the other cases in the case base. In simple terms, a case that matches the query case on n number of features, will be retrieved rather than a case which matches on k number of features where $k < n$. Some features that are considered more important in a problem solving situation may have their importance denoted by weighting these features more heavily in the matching.

3.1.2 Inductive approaches

When inductive approaches are used to determine the case base structure, that is to determine the relative importance of features for discriminating between similar cases, the resulting hierarchical structure of the case base provides a reduced search space for the case-retriever. This may in turn reduce the search time for queries.

3.1.3 Knowledge Guided Approaches

Knowledge guided approaches to retrieval use domain knowledge to determine the features of a case which are important for that case in particular to be retrieved in future. In some situations different features of each case will have been important for the success level of that case.

As with the inductive approaches to retrieval, knowledge guided indexing may result in a hierarchical structure, effective for searching.

3.1.4 Validated Retrieval

There have been numerous attempts at improving these forms of retrieval. Validated Retrieval proposed by Simoudis is one of these[12].

Validated Retrieval consists of two phases, firstly the retrieval of *all* cases that appear to be relevant to a problem, based on the main features of the query case.

The second phase involves deriving more discriminating features from the group of retrieved cases to determine whether they(the cases) are valid in the current situation. The advantage of this method is that inexpensive methods can be used to make the initial retrieval from the case base, while more expensive methods can be used in the second phase as they are applied to only a subset of the case base.

This is just one of many possible alternatives, for retrieval. There are a number of factors therefore to consider when determining the method of retrieval.

- The number of cases to be searched
- The amount of domain knowledge available
- The ease of determining weightings for individual features
- Whether cases should be indexed by the same features or whether each case may have varying important features.

Once a case has been retrieved there is usually a phase to determine whether a case is close enough to the problem case or whether the search parameters need to be modified and the search conducted again. There can be a significant time saving if the right choice is made. The adaptation time for a distant case could be significantly greater than searching again. When considering an analysis method for this determination, the following points should be considered:

- The time and resources required for adaptation
- The number of cases in the case base, i.e. how likely is it that there is a closer case.
- The time and resources required for search
- How much of the case base has already been searched in previous pass(es).

If we now look at the processes involved in CBR this far we can represent these succinctly as shown in Figure 4:

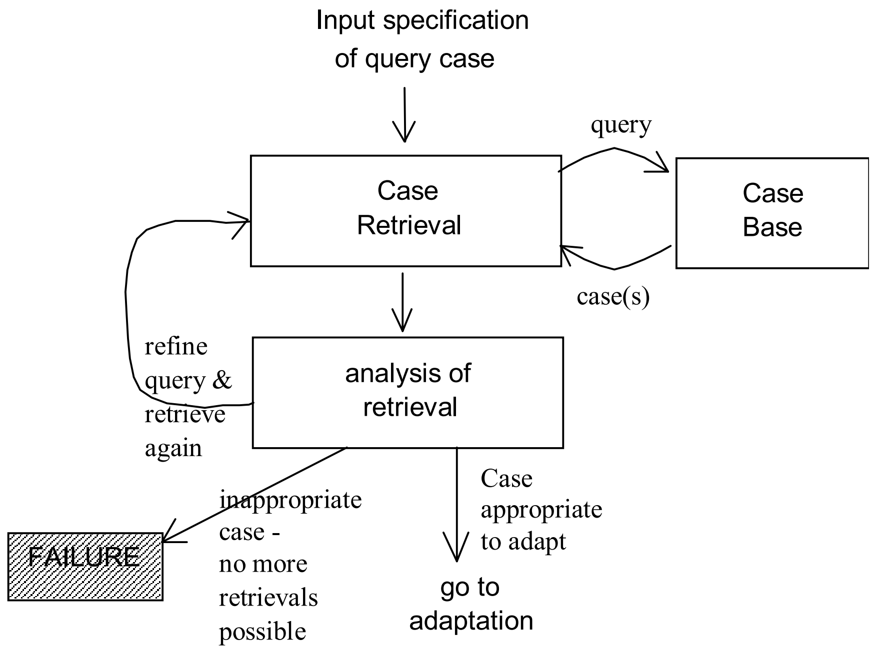

Figure 4. Process involved in CBR

3.2 Case Adaptation

Case adaptation is the process of translating the retrieved solution into the solution appropriate for the current problem. It has been argued that adaptation may be the most important step of CBR as it adds intelligence to what would otherwise be simple pattern matchers [13].

There are a number of approaches that can be taken to carry out case adaptation.

- The solution returned could be used as a solution to the current problem, without modification, or with modifications where the stored solution is not entirely appropriate for the current situation.
- The steps or processes that were followed to obtain the previous solution could be re-run, without modifications, or with modifications where the steps taken in the past solution are not fully satisfactory in the current situation.
- Where more than one case has been retrieved a solution could be derived from multiple cases, or alternatively several alternative solutions could be presented.

Adaptation can use various techniques, including rules or further case-based reasoning on the finer grained aspects of the case. When choosing a strategy for case adaptation it can be helpful to consider the following :

- On average how close will the retrieved case be to the query case?

- How many characteristics will differ between the cases in the usual situation
- Are there commonsense or otherwise known rules that can be applied to do the adaptation

After adaptation has been carried out it is desirable to check that the adapted solution takes into account the differences between the retrieved case and the current problem. That is, that adaptation has addressed the differences between them. There is a need to consider here what action is to be taken in the event that this check determines that the proposed solution is unlikely to be successful.

At this stage the developed solution is ready for testing/use in the applicable domain. This stage concludes the necessary steps for all CBR systems, however many systems will now enter a learning phase as shown in Figure 5.

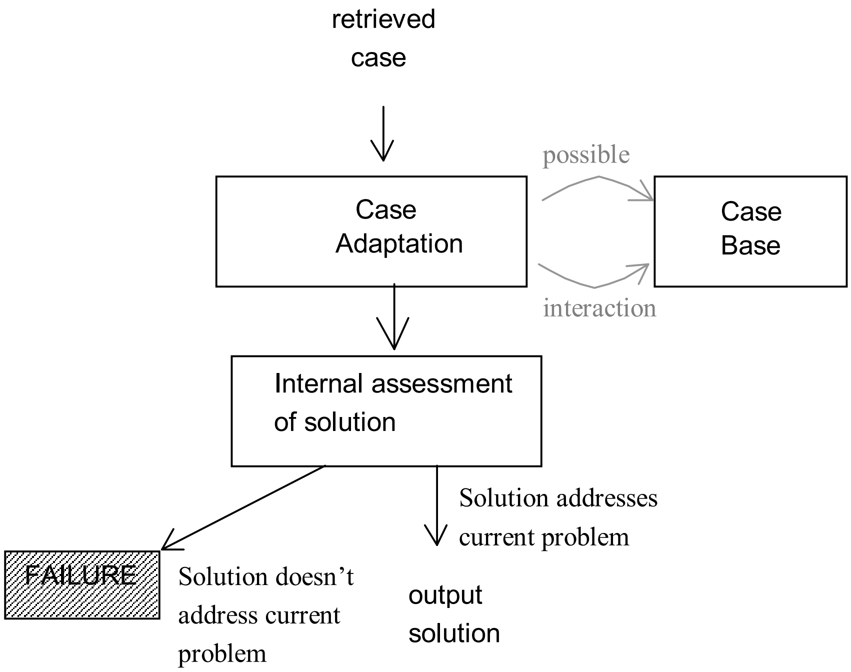

Figure 5. CBR enter into a learning state

4 Learning and Maintenance

4.1 Learning in CBR systems

Once an appropriate solution has been generated and output, there is some expectation that the solution will be tested in reality. To test a solution we have to consider both the way it may be tested but also how the outcome of the test will be classified as a success or a failure. In other words some criteria need to be defined for the performance rating of the proffered solution.

Using this real world assessment the CBR system can be updated to take into account any new information uncovered in the processing of the new solution. This information can be added to the system for two purposes. Firstly the more information that is stored in the case base, the closer the match found in the case base is likely to be. The second purpose of adding information to the case base is for is to improve the solution the CBR is able to create.

Learning may occur in a number of ways. The addition of the new problem, solution and outcome to the case base is a common method. The addition of cases to the case base will increase the range of situations covered by the cases and reduce the average distance between an input vector and the closest stored vector as shown in Figure 6.

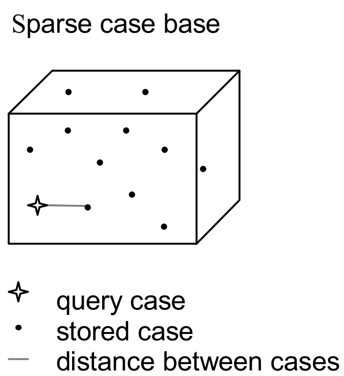

Figure 6. Distance between cases

A second method of learning in the CBR system is using the solution's assessment to modify the indices of the stored cases or to modify the criteria for case retrieval. If a case has indices that are not relevant to the contexts they should be retrieved in, adjusting these indices may increase the correlation between the times a case is retrieved and the times a case ought to be retrieved.

Likewise, the assessment of the solution's performance may lead to an improved understanding of the underlying causal model of the domain which can be used to the improved adaptation processing. If better ways to modify the cases with respect to the distance between the query and retrieved cases can be found, the output solution will be likely to be improved.

In the event that learning is occurring by way of the addition of new cases to the case base there are a number of considerations:

In which situations should a case be added to the case base and in which situations should it be discarded?
To determine this we have to consider the level of success of the solution, how similar it is to other cases in the case base and whether there are important lessons to be learned from the case.

If the case is to be added to the case base, both the indices of the new case must be determined and how that case is to be added to the case base.
If the case base's structure and retrieval method are highly structured, for example an inductively determined hierarchical structure or a set of neural networks, the incorporation of a new case may require significant planning and restructuring of the case base.

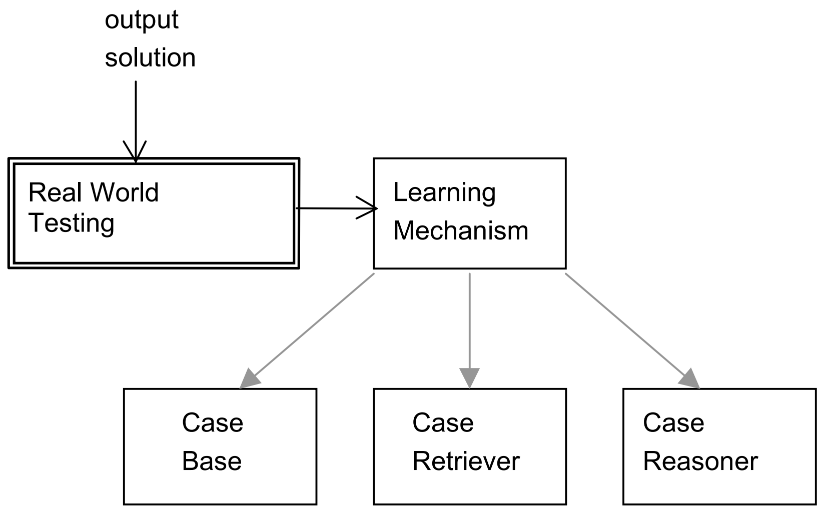

Figure 7. Learning Mechanism in CBR

4.2 CBR maintenance

When applying CBR systems for problem solving, there is always a tradeoff between the number of cases to be stored in the case library and the retrieval efficiency. The larger the case library, the more the problem space covered, however, it would also downgrade the system performance if the number of cases grows to an unacceptable high level. Therefore, removing the redundant cases or less useful cases under an acceptable error-level is one of the most important tasks to maintain CBR systems. Leake and Wilson [14] defined case-base maintenance as the implementation of policies for revising the organization or contents (representation, domain content, accounting information, or implementation) of the case-base in order to facilitate future reasoning for a particular set of performance objectives.

The central idea for CBR maintenance is to develop some measures for case competence, that is the range of problems the CBR can solve. There are various properties that may be useful, such as the size, the distribution and the density of cases; the coverage of individual cases; the similarity and adaptation knowledge of a given system [15]. Coverage refers to the set of problems that each case could solve whilst reachability refers to the set of cases that could provide solutions to the current problem [16]. The higher the density of cases, the chances of having redundant cases increase. By expressing the density as a function of case similarity, deletion policy could be formulated for removing cases which are highly reachable by others.

Another need for CBR maintenance is the possible existence of conflicting cases in the case library due to changes of domain knowledge or task environments. For examples, when more powerful cases exist which may contain inconsistent information either with other parts of the same case or with other more primitive original cases. Furthermore, if two cases are considered equivalent (with identical feature values) or one case subsumes another by having more feature criteria, then a maintenance process may be required to remove these redundant cases.

5 The role of soft computing in CBR

Increasingly, CBR is now being recognized as an effective problem solving methodology that constitutes a number of phases, i.e. case representation, indexing, similarity comparison, retrieval and adaptation. For complicated real world applications, some degree of fuzziness and uncertainty is always encountered, soft computing techniques, such as fuzzy logic, neural networks and genetic algorithms will be very useful in areas where uncertainty, learning or knowledge inference are parts of the system's requirement. In order for us to gain an understanding of these techniques so as to identify approaches for their used in CBR, we briefly summarized them in the sections below:

5.1 Fuzzy Logic

Fuzzy theory has been successfully applied to computing with words or matching of linguistic terms for reasoning. In the context of CBR, when quantitative features are used to create indexes, it involves the conversion of the numerical features into qualitative terms for indexing and retrieval. These terms are always fuzzy terms. Moreover, one of the major issues in fuzzy theory is about measuring similarities for designing robust systems. The notion of similarity measurement in CBR is also fuzzy in nature. For example, Euclidean distances of features are always used to represent the similarity among cases, however the use of fuzzy theory for indexing and retrieval has many advantages [17] over crisp measurements such as the following:

- Numerical features could be converted to fuzzy terms to simplify comparison
- Fuzzy sets allow multiple indexing of a case on a single feature with different degrees of membership
- Fuzzy sets make it easier to transfer knowledge across domains
- Fuzzy sets allow term modifiers to be used to increase the flexibility in case retrieval

Other application of fuzzy logic to CBR includes the use of fuzzy production rules to guide case adaptations. For example, fuzzy production rules may be discovered from a case library to associate the similarity between problem features and solution features of cases.

5.2 Neural Networks

Artificial Neural Networks (ANNs) are usually used for learning and generalization of knowledge and patterns. They are not appropriate for expert reasoning and their explanation abilities are extremely weak. Therefore, many applications of ANNs in CBR systems tend to employ a loosely integrated approach where the separate ANN components have some specific objectives such as classification and pattern matching. The benefits of using neural networks for retrieving cases include the following: essentially case retrieval is the matching of patterns, a current input pattern (case) with one or more stored patterns or cases. Neural networks are very good at matching patterns. They cope very well with incomplete data and the imprecision of inputs, which is of benefit in the many domains, as sometimes some portion is important for a new case while some other part is of little relevance. Domains that use the case-based reasoning technique are usually complex. This means that the classification of cases at each level is normally non-linear and hence that for each classification a single-layer network is not sufficient and a multi-layered network is required.

Hybrid CBR and ANNs are very common architecture for complicated applications. Knowledge may first be extracted from the ANNs and represented by symbolic structures for later use by other CBR components. On the other hand, ANNs could be used for retrieval of cases where each output neuron represents one case.

5.3 Genetic Algorithms

Genetic algorithms (GA) are adaptive techniques that are used to solve search and optimization problems inspired by the biological principles of natural selection and genetics. In GA, each individual is represented as a string of binary values, populations of competing individuals evolve over many generations according to some fitness function. A new generation is produced by selecting the best individuals and mating them to produce a new set of offspring. After many generations, the offspring will bear all the most promising characteristics, and will be adopted as the potential solution for the search problem. Learning local and global weights of case features is one of the most popular applications of GA to CBR. The discovery or learning of these weights will indicate how important the features within a case are with respect to the solution features. This weights information can improve the design of retrieval accuracy of CBR systems.

6 Application Areas for CBR

Case-based reasoning has been applied in many different areas as the systems mentioned here will testify. A look at what they have done shows the versatility of the paradigm and also provides an insight into the directions artificial intelligence research can take. The domains seem limitless and the following systems are just a fraction of the research and commercial systems in existence.

6.1 Law

Hypo is an adversarial case-based reasoning system that deals with trade secrets law[18]. Probably the best known and most documented of all case-based reasoning systems, Hypo was developed by Kevin Ashley and Edwina Rissland at the University of Massachusetts. Hypo analyses problem situations in the trade law area and retrieves relevant cases from its case base, forms them into legal arguments

Kowalski's System, The Malicious Prosecution Consultant (MPC)[19] is a CBR system that operates in the domain of malicious prosecution.

HELIC-II[20] (Hypothetical Explanation constructor by Legal Inference with Cases by 2 inference engines) is a hybrid legal system for the penal code, using legal rules (the law) and cases (precedents).

William Bain, in his PhD dissertation[21], discusses the system JUDGE. JUDGE is a case based reasoning system that attempts to model the sentencing of criminals done by real-life judges by comparing current cases to what judges have done in the past. He used interviews with judges asking them to describe what they would do in certain hypothetical cases to determine the factors involved.

Zeleznikow et al.[22] have developed a system called IKBALS operating in the field of workers compensation law. This is implemented as a hybrid rule based and case based system.

Another System is OPINE[23] a generic case-based reasoner for use in legal domains. OPINE is different to the previously described CBR systems as it has only a single function and that is to provide evaluation of likely case outcome.

An earlier system by Lambert & Grunewald[24] is LESTER (Legal Expert System for Termination of Employment Review), a case-based reasoning program in the area of unjust discharge from employment under collective bargaining agreements.

A fuzzy CBR system for legal inference is developed by Hirota et. at. [25]. This implemented fuzzy features and case rules for contract legal cases.

Another system which used ANNs for knowledge extraction is HILDA [26], which use the knowledge extracted from a ANN to guide the rule inferences and cases retrieval.

Hollatz [27] also developed a neuro-fuzzy approach in legal reasoning which tried to find structure in precedent decisions as well as to identify legal precedents.

6.2 Medicine

CASEY[28][29] is a program designed by Koton at MIT for her PhD. dissertation. It analyses descriptions of patients with heart disease and to produce a diagnostic explanation of the symptoms of the patients condition. CASEY integrates case-based and causal reasoning with a model-based expert system

Protos[30] is an exemplar-based learning apprentice. It is not domain specific but it has been applied in the field of clinical audiology. It is used to classify cases into categories and to find an exemplar that closely matches the current input case.

BOLERO[31][32], designed by Lopez and Plaza diagnoses the cause of pneumonia in patients so they can be treated. It learns not only from its successes but also its failures.

Kolodner and Kolodner[33]discuss the role of experience in diagnosing psychological problems and mention their system, SHRINK, which models a small part of this problem.

Hsu [34] developed a hybrid case-based system to help physician. It uses a distributed fuzzy neural network for case retrieval, and other decision support techniques for selecting and adapting relevant cases from the case library for assisting medical consultation.

6.3 Engineering

Archie[35] is a case-based design support tool, for use in the architectural design of office buildings. It gives architects a case base of architectural designs created by other architects, and aids in determining factors that solved problems in past designs. It is used in the high-level conceptual design of buildings, rather than at the drafting or engineering stage.

CADSYN, developed by Maher & Zhang[36] is a hybrid case base designer in the field of structural design. Case Based Reasoning is combined with decomposition. Ie. a design is composed of a number of parts. To adapt a design it can be broken down into smaller adaptation problems, so that part of a design is transformed at one time.

On a different tangent is GENCAD[37] applied to the layout of residential buildings so that they conform to the principles of feng shui.

6.4 Computing - Help Desk Support

Cascade[12] is a case-based system that uses validated retrieval. Its purpose is to aid help-desk engineers find solutions to resolve device driver failures affecting Digital Equipment Corporation's VMS operating system. Device driver failure causes 60 percent of VMS crashes.

Kriegsman & Barletta[38] have built a system using ReMind, Cognitive System's Case-Based reasoner building tool for creating a CBR to utilise the logs at General Electric's Help Desk to provide solutions to the constant stream of problems their Help Desk operators attempt to solve.

6.5 Communication networks

CRITTER[39][40][41] is a case-based reasoning trouble ticketing system for managing and resolving network faults in computer/telecommunication systems. In trouble ticketing systems, when a problem is detected a ticket is created which is kept until the problem is resolved. The ticket includes information describing the tickets problem and it's resolution when solved. Therefore in this system tickets are used as the past cases in the case base and from these new solutions are created.

Schenker [42] combines fuzzy logic and case-based reasoning to determine fault-prone modules in a communication network.

6.6 Manufacturing design

Clavier[43] is a case-based planning system that aims to help autoclave loaders design successful autoclave loadings. An autoclave is a large pressurized convection oven that is used to cure graphite threaded composite materials. Autoclave operators work with a list of prioritized parts and try to create a loading configuration that includes the most number of high-priority parts. It is more difficult than it seems as all the parts must heat up at approximately the same rate, and this is affected by a number of factors.

Main and Dillon[44][45][46] developed a system for fashion foot design, using a fuzzy approach to case representation and neural networks for retrieval.

Klinger et al.[47] developed a system, Bidder's Associate, which assists in the preparation of bids for manufactured parts.

6.7 Finance

Morris describes a system in this area called SCAN[48]. It is a CBR system in the field of information systems auditing and is designed to help the inexperienced auditor in evaluating controls and proposing audit recommendations. Cases in SCAN are traces of past audit cases, and they are indexed and retrieved in a traditional CBR fashion.

Jo [49] has integrated CBR, ANNs and Discriminant Analysis for Bankruptcy Prediction. The prediction ability of the integrated model is superior to the three independent predication techniques.

6.8 Job Shop Scheduling

Miyashita & Sycara[50] use CBR in their system CABINS, a system which uses a case-based learning method for acquiring context-dependent user optimization preferences and tradeoffs and using them to incrementally improve schedule quality in predictive scheduling and reactive schedule management in response to unexpected execution events. Cases in this system are used in three ways: for repair action selection; evaluation of the intermediate repair results; and in recovery from revision failures.

Kettler et al.[51] are using their CAPER (parallel Retrieval) system in the car assembly domain. Kettler et. al. use the parallelism of the connection machine to retrieve cases and plans from a large unindexed memory. Their system can retrieve cases and plans based on any feature of the target problem, including abstractions of target features.

6.9 Scheduling

Koton's SMARTplan[52] is used to schedule the tasks involved in and allocate resources for large scale airlift operations which involve thousands of individual tasks. The requirements for a plan involve requests to move personnel and cargo from one location to another within a specified time window. The resulting plan consists of aircraft allocations, routing networks and airfield assignments.

Miyashita [53] proposed an integrated architecture for distributed planning and scheduling that exploits constraints for problem decomposition, coordination and case-based reasoning.

6.10 Language

The problem MBRTALK[54] solves is the pronunciation of novel words and the cases it contains in its case base are words with their phonetic pronunciation (defined in combination of structures). MBRTALK firstly calculates the dissimilarity between the current letter of the word to be pronounced and those letters in the case base. It retrieves n of the most closely matching cases and if the pronunciations of the target letter is the same for all n cases, then a pronunciation is predicted, otherwise, the possible pronunciations are ordered in terms of which is the most likely to be correct..

There is also another system called PRO[55][56]. PRO differs from MBRTALK in the environment it runs in and hence in the memory structure it uses. PRO also differs a little in how it solves a problem of word pronunciation. It first looks up a hypothesis base to find hypotheses for the pronunciation of a certain word and the uses it's case base and statistical base to decide which of these hypotheses is most likely. Thus both systems for word pronunciation, MBRTALK and PRO fall into the statistically based paradigm, and both use large case bases to find the most likely pronunciations of letters and combinations of letters within a case base.

6.11 Explanation and understanding of Stories

Kass's ABE[57][58] (Adaptation-Based Explanation) project develops explanations for events in stories.

Kass et al.[59] describe SWALE which had to understand a story - that of Swale , a three year old racehorse who died suddenly. TWEAKER follows on from the SWALE system and focuses on the tasks of explanation application and adaptation. TWEAKER is designed to explain deaths and disasters.

6.12 Food

Chef[60] is a case-based planner in the domain of Szechwan cooking and its purpose is to create new recipes on the basis of the user's request(s). It builds plans out of its memory of old ones and the approach it takes is to anticipate problems and avoid them.

Kolodner's JULIA program[61][62][63][64] is used in meal planning, how to plan a meal based on the guests preferences, to cater for people who are vegetarians or don't like fish etc. There is another system JULIANA by Hong Shinn one of Kolodner's students. JULIANA plans meals for institutions such as schools or nursing homes.[64]

6.13 *Route Finding*

Liu et. al.[65][66] developed a system for finding routes in Singapore called R-Finder. This system combines Dijkstra's Algorithm to search for shortest routes, Knowledge Approaches to reduce the area to be searched, and case based reasoning to remember whether it has a route stored in memory the same as the one required or one to very close places that can be adapted.

Goel et al.[67] have also implemented a number of systems for route planning: ROUTER1, ROUTER2, and ROUTER3. ROUTER2 uses case-based reasoning while ROUTER3 uses case-based reasoning in combination with model-based reasoning. These systems are involved in Route planning around the Georgia Tech Campus

Kettler et al.[51] are using their CAPER (parallel retrieval system in the transportation logistics domain.

6.14 *Materials Handling*

Li and Dahan[68] developed a system COESDOES (Case-Oriented Expert system for the Design of Orienting Systems) which designs orienting systems which are used to feed components for automatic assembly. These can be systems such as automatic vibratory and centrifugal feeders, and their design is time consuming.

6.15 *Telephone Demand.*

Lee et. al.[69] created an expert system using CBR for forecasting irregular telephone demand which will occur in specific areas by region development.

Kopeikina et al.[70] describe a system for continuous management of traffic in the standard public switched telephone network which involves allocating a changing set of network resources to satisfy demands from a fluctuating pattern of calls.

6.16 *The Environment*

AIRQUAP[71] is a system that is used to predict the level of a particular air pollutant in Athens, Greece. The pollutant it predicts the level of is NO_2 , a secondary photochemical pollutant which is one of the hardest to predict and has high concentration levels in Athens and other cities with similar climates. It uses data collected over previous years to predict by 9 am what the level of NO_2 will be that day.

Krovvidy[72] et al developed a system for wastewater treatment using case based reasoning. The task in such a system is to determine a treatment train of processes to be performed on wastewater to lower the impurity levels to the acceptable ranges

6.17 Fault Diagnosis.

Karamouzis & Feycock[73] have developed a system called EPAION which combines CBR with MBR and is being tested in the domain of in-flight fault diagnosis and prognosis of aviation subsystems, particularly jet engines.

Liu and Yan [74] have developed a CBR system using fuzzy logic type neural networks for diagnosing electronic systems.

7 Recapitulation

In this tutorial, we gave a brief explanation of case based reasoning, its main components, advantages as well as the situations when it is most useful. We next briefly outlined some of the most common soft computing techniques and their relevance to case based reasoning. Lastly we provided a birdseye view of some of the most important applications. There are many important books and monographs the interested reader should follow up and these include but are not limited to

- Case-Based Reasoning Research and Development, Proceedings of the Third International Conference on Case-based Reasoning, (Eds) Klaus-Dieter Althoff, Ralph Bergmann and L. Karl Branting, ICCBR-99, Seeon Monastery, Germany, July 1999.
- Applying Case-Based Reasoning: Techniques for Enterprise Systems, by Ian Watson, Morgan Kaufmann Publishers, Inc., 1997.
- Case-Based Reasoning Experiences, Lessons and Future Directions, (Ed) David B. Leake, AAAI Press/MIT Press, 1996.
- Case-Based Reasoning, by Janet Kolodner, Morgan Kaufmann, San Mateo, CA, 1993.

The following two web sites provide many useful information and links to other CBR resources:

- Ian Watson at the University of Salford maintains a site with URL http://www.ai-cbr.org
- Ralph Bergmann, Ivo Vollrath and Sascha Schmitt at the University of Kaiserslautern maintain a site with URL http://www.cbr-web.org

8 References

[1] Janet Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA, 1993.

[2] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1):3-34, 1992.

[3] Janet Kolodner and William Mark. Case-based reasoning. *IEEE Expert*, 7(5):5-6, October 1992.

[4] Kevin D. Ashley. Case-based reasoning and its implications for legal expert systems. *Artificial Intelligence and Law: An International Journal,* 1(2-3):113-208, 1992.
[5] Gebhardt, F., Vob, A., Grather, W., Schmidt-Beltz, B., Reasoning with Complex Cases, *Kluwer Academic Publishers*, Norwell, MA (1997).
[6] Birnbaum, L and Collings, G, 1989, “Remindings and engineering design themes: a case study in indexing vocabulary”, in *Proceedings of second Workshop on Case-based Reasoning*. Pensacola Beach, FL., 1989.
[7] Hammond, K.J. 1989, “On functionally motivated vocabularies: an apologia,” In *Proceedings second Workshop on Case-based Reasoning*, Pensacola Beach, FL, 1989.
[8] Andrea Bonzano, Padraig Cunningham and Barry Smyth, "Using Introspective Learning to Improve Retrieval in CBR: A Case Study in Air Traffic Control," in *Proceedings of second International Conference on Case-based Reasoning*, RI, USA, 1997, pp. 291-302.
[9] Stefanie Bruninghaus and Kevin D. Ashley, "Using Machine Learning for Assigning Indices to Textual Cases," in *Proceedings of second International Conference on Case-based Reasoning*, RI, USA, 1997, pp. 303-314.
[10] Acorn, T and Walden, S., 1992 "SMART: Support management cultivated reasoning technology Compaq customer service," in *proceeding of AAAI-92*, AAAI Press/MIT Press.
[11] Barry Smyth and Mark T. Keane, "Adaptation-guided retrieval: questioning the similarity assumption in reasoning," *Artificial Intelligence*, 102 (1998) pp.249-293.
[12] Evangelos Simoudis. Using case-based retrieval for customer technical support. *IEEE Expert*, 7(5):7-12, October 1992.
[13] Mehmet Adalier and Costas Tsatsoulis. Redesigning for manufacturability using REINRED. *Applied Artificial Intelligence*, 6(3):285-302, July-Sept. 1992.
[14] David B. Leake and David C. Wilson, "Categorizing Case-Base Maintenance: Dimensions and Directions," in *Proceedings of 4th European Workshop*, EWCBR-98, 1998, pp.197-207.
[15] Barry Smyth and Elizabeth McKenna, "Modelling the Competence of Case-Bases," in *Proceedings of 4th European Workshop*, EWCBR-98, 1998, pp.208-220.
[16] Kirsti Racine and Qiang Yang, "Maintaining Unstructured Case Bases," in *Proceedings of second International Conference on Case-based Reasoning*, RI, USA, 1997, pp. 553-564.
[17] Bing Chiang Jeng and Ting Peng Liang, "Fuzzy Indexing and Retrieval in Case-Based Systems," *Expert Systems with Applications*, Vol.8, No.1, pp.135-142, 1995.
[18] Kevin D. Ashley. *Modeling legal Argument: Reasoning with Cases & Hypotheticals.* MIT Press, Cambridge, MA, 1990.
[19] Andrzej Kowalski. Case-based reasoning and the deep structure approach to knowledge representation. In *Proceedings of the Third International Conference on Artificial Intelligence and Law:*

Oxford, England: 25-28 June 1991[77], pages 21-30.

[20] Katsumi Nitta, Yoshihisa Ohtake, Shigeru Maeda, Masayuki Ono, Hiroshi Ohsaki, and Kiyokazu Sakane. HELIC-II: a legal reasoning system on the parallel inference machine. In *FGCS '92: Proceedings of the International Conference on Fifth Generation Computer Systems: Tokyo, Japan: 1-5 June 1992*, volume 2, pages 1115-1124, Burke, VA, 1992. ICOT, IOS Press.

[21] William Michael Bain. *Case-Based Reasoning: A Computer Model of Subjective Assessment.* PhD thesis, Department of Computer Science, Yale University, 1986.

[22] John Zeleznikow, George Vossos, and Daniel Hunter. The IKBALS project: Multi-modal reasoning in legal knowledge based systems. *Artificial Intelligence and Law*, 2:169-203, 1994.

[23] Kenneth A. Lambert and Mark H. Grunewald. Legal theory and case-based reasoners: The importance of context and the process of focusing. In *Proceedings of the Third International Conference on Artificial Intelligence and Law: Oxford, England: 25-28 June 1991* [77], pages 191-195.

[24] Kenneth A. Lambert and Mark H. Grunewald. LESTER: Using paradigm cases in a quasi-precedential legal domain. In *Proceedings of the Second International Conference on Artificial Intelligence and Law: Vancouver, BC, Canada: 1989*, pages 87-92, New York, NY, 1989. International Conference on Artificial Intelligence and Law, ACM Press.

[25] Hirota, K., Yoshino, H., Xu, M.Q., Zhu, Y., Horie, D., "A fuzzy case based reasoning system for the legal inference," *in Proceedings of fuzzy systems*, 1998, World Congress on Computational Intelligence, pp.1350-1354.

[26] Egri, P.A., Underwood, P.F., "HILDA: Knowledge extraction from neural networks in legal rule based and case-based reasoning," *Neural Networks, 1995 Proceedings*, IEEE international conference on, Vol. 4. Pp. 1800-1805.

[27] Hollatz, J., "Neuro-fuzzy in legal reasoning," in *Proceedings in Fuzzy Systems, International Joint conference of the 4th IEEE International Conference on Fuzzy Systems and the 2nd International Fuzzy Engineering Symposium*, pp.655-662.

[28] Phyllis Koton. Reasoning about evidence in causal explanation. In *AAAI-88: Proceedings of the Seventh National Conference on Artificial Intelligence: Saint Paul, MN: 1988,* volume 1, pages 256-261, Palo Alto, CA, 1988. American Association for Artificial Intelligence, Morgan Kaufmann.

[29] Phyllis Koton. A medical reasoning program that improves with experience. *Computer Methods and Programs in Biomedicine*, 30(2-3):177-184, Oct-Nov 1989.

[30] Ray Bareiss. *Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification and Learning.* Academic Press, Boston, MA, 1989.

[31] Beatriz Lopez and Enric Plaza. Case-based learning of strategic knowledge. In Y. Kodratoff, editor, *Machine Learning - EWSL-91:*

European Working Session on Learning Proceedings: Porto, Portugal. 6-8 March 1991, pages 398-411, Berlin, Germany, 1991. Springer-Verlag.

[32] Beatriz Lopez and Enric Plaza. Case-based planning for medical diagnosis. In J. Komorowski and Z. W. Ras, editors*, ISMIS'93: Methodologies for Intelligent Systems. Proceedings of the 7th International Symposium: Trondheim, Norway: 15-18 June 1993*, pages 96-105, Berlin, Germany, 1993. Springer-Verlag.

[33] Janet L. Kolodner and Robert M. Kolodner. Using experience in clinical problem solving: Introduction and framework. *IEEE Transactions on Systems, Man and Cybernetics,* 17(3):420-431, May/June 1987.

[34] Chien Chang Hsu and Cheng Seen Ho, "A hybrid case-based medical diagnosis system," *Tools with Artificial Intelligence*, 1998 Proceedings, Tenth IEEE International conference on, pp. 359-366.

[35] Ashok K. Goel, Janet L. Kolodner, Michael Pearce, Richard Billington, and Craig Zimring. Towards a case-based tool for aiding conceptual design problem. In E. R. Bareiss, editor, *Proceedings of a DARPA Workshop on Case-Based Reasoning: Washington, DC: 1991,* pages 109-120, San Mateo, CA, 1991. DARPA, Morgan Kaufmann.

[36] Mary Lou Maher and Dong Mei Zhang. Cadsyn: a case-based design process model. *AI EDAM: Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 7(2):97-110, 1993.

[37] Andrés Gómez de Silva Garza and Mary Lou Maher. An evolutionary approach to case adaptation. In Althoff et al. [75] pages 162-172.

[38] Mark Kriegsman and Ralph Barletta. Building a case-based help desk application. *IEEE Expert*, 8(6):18-26, December 1993.

[39] Lundy Lewis. A case-based reasoning approach to the management of faults in communications networks. In *IEEE INFOCOM '93: Proceedings of the Conference on Computer Communications. Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the future: San Fransisco, CA: 28 March - 1 April 1993,* volume 3, pages 1422-1429. IEEE Computer and Communications Societies, IEEE Comput. Soc. Press, 1993.

[40] Lundy Lewis. A case-based reasoning approach to the management of faults in communications networks. In *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications: Orlando, FL: 1-5 March 1993*, pages 114-119, Los Alamitos, CA, 1993. IEEE, IEEE Comput. Soc. Press.

[41] Lundy Lewis. A case-based reasoning approach to the resolution of faults in communications networks. *IFIP Transactions C [Communication Systems],* C-12, Integrated Network Management, III:671-682, 1993.

[42] Schenker, D.F., and Khoshgoftaar, T.M., "The application of fuzzy enhanced case-based reasoning for identifying fault-prone

modules," in *Proceedings of 1998 Third IEEE International Conference on High-Assurance Systems Engineering Symposium*, pp. 90-97.

[43] William S. Mark. Case-based reasoning for autoclave management. In *Proceedings of a DARPA Workshop on Case-Based Reasoning: Pensacola Beach, Florida: 1989* [79], pages 176-180.

[44] Julie Main, Tharam S. Dillon, and Rajiv Khosla. Use of neural networks for case-retrieval in a system for fashion shoe design. In Graham F. Forsyth and Moonis Ali, editors, *IEA/AIE 95: Industrial and Engineering Applications of Artificial Intelligence and Expert Systems: Proceedings of the Eighth International Conference: Melbourne, Australia: June 6-8 1995,* pages 151-158, USA, 1995. Gordon and Breach.

[45] Julie Main, Tharam S. Dillon, and Rajiv Khosla. Use of fuzzy feature vectors and neural networks for case retrieval in case based systems. In M. H. Smith, M. A. Lee, J. Keller, and J. Yen, editors, *NAFIPS: 1996 Biennial Conference of the North American Fuzzy Information Processing Society: Berkeley, CA: 19-22 June 1996*, pages 438-443, New York, NY, 1996. IEEE.

[46] Julie Main and Tharam S. Dillon. A hybrid case-based reasoner for footwear design. In Althoff et al. [75], pages 499-509.

[47] David W. Klinger, Richard Stottler, and Steven R. LeClair. Manufacturing application of case-based reasoning. In *NAECON 1992: Proceedings of the IEEE 1992 National Aerospace and Electronics Conference: Dayton, OH: 18-22 May 1992,* volume 3, pages 855-859, New York, NY, 1992. IEEE.

[48] Bonnie W. Morris. SCAN: A case-based reasoning model for generating information system control recommendations. *International Journal of Intelligent Systems in Accounting, Finance and Management*, 3(1):47-63, January 1994.

[49] Hongkyu Jo and Ingoo Han, "Integration of Case-based Forecasting, Neural Network, and Discriminant Analysis for Bankruptcy Prediction," *Expert Systems with Applications*, Vol. 11, No.4, pp. 415-422, 1996.

[50] Kazuo Miyashita and Katia Sycara. CABINS: a framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair. *Artificial Intelligence*, 76(1-2):377-426, July 1995.

[51] Brian P. Kettler, James A. Hendler, William A. Anderson, and Matthew P. Evett. Massively parallel support for case-based planning. *IEEE Expert*, 9(1):8-14, February 1994.

[52] Phyllis Koton. SMARTplan: A case-based resource allocation and scheduling system. In *Proceedings of a DARPA Workshop on Case-Based Reasoning: Pensacola Beach, Florida: 1989* [79], pages 285-289.

[53] Miyashita, K. and Hori, M., "A distributed architecture for planning and scheduling that learns through negotiation cases," in 1996 Proceedings of IEEE conference on Emerging Technologies and Factory Automation, pp.136-142.

[54] Craig W. Stanfill. Memory-based reasoning applied to English pronunciation. In *AAAI-87: Proceedings of the Sixth National Conference on Artificial Intelligence: Seattle, WA: 1987* [76], pages 577-581.

[55] Wendy G. Lehnert. Case-based problem solving with a large knowledge base of learned cases. In *AAAI-87: Proceedings of the Sixth National Conference on Artificial Intelligence: Seattle, WA: 1987* [76], pages 301-306.

[56] Wendy G. Lehnert. Word pronunciation as a problem in case-based reasoning. In *Program of the Ninth Annual Conference of the Cognitive Science Society: Seattle, Washington: 1987,*[78], pages 120-130.

[57] Alex Kass. Adaptation-based explanation: Explanation as cases. In *6th International Workshop on Machine Learning: Ithaca, NY: 1989*, pages 49-51, San Mateo, CA, 1989. International Workshops on Machine Learning, Morgan Kaufmann.

[58] Alex Kass. Adaptation-based explanation: Extending script/frame theory to handle novel input. In *IJCAI 89: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence: Detroit, MI: 20-25 August 1989,* volume 1, pages 141-147, San Mateo, CA, 1989. International Joint Conferences on Artificial Intelligence, Morgan Kaufmann.

[59] Alex M. Kass and David B. Leake. Case-based reasoning applied to constructing explanations. In Kolodner [80], pages 190-208.

[60] Kristian J. Hammond *Case-Based Planning: Viewing Planning as a Memory Task.* Academic Press, Boston, MA, 1989.

[61] Janet L. Kolodner. Capitalizing on failure through case-based inference. In *Program of the Ninth Annual Conference of the Cognitive Science Society: Seattle, Washington: 1987* [78], pages 715-726.

[62] Janet L. Kolodner. Extending problem solver capabilities through case-based inference. In *4th International Workshop on Machine Learning: Irvine: 1987,* pages 167-178, San Mateo, CA, 1987. International Workshops on Machine Learning, Morgan Kaufmann.

[63] Janet L. Extending problem solver capabilities through case-based inference. In Kolodner [80], pages 21-30.

[64] Janet L. Kolodner. Understanding creativity: A case-based approach. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *EWCBR-93: Topics in Case-Based Reasoning. First European Workshop, Selected Papers: Kaiserslautern, Germany: November 1993*, pages 3-20, Berlin, Germany, 1993. Springer-Verlag.

[65] Bing Liu, Siew-Hwee Choo, Shee-Ling Lok, Sing-Meng Leong, Soo-Chee Lee, Foong-Ping Poon, and Hwee-Har Tan. Finding the shortest route using cases, knowledge, and Dijkstra's algorithm. *IEEE Expert*, 9(5):7-11, October 1994.

[66] Bing Liu, Swie-Hwee Choo, Shee-Ling Lok, Sing-Meng Leong, Soo-Chee Lee, Foong-Ping Poon, and Hwee-Har Tan. Integrating case-based reasoning, knowledge-based approach and Dijkstra

algorithm for route finding. In *Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Applications: San Antonio, TX: 1-4 March 1994*, pages 149-155, Los Alamitos, CA, 1994. IEEE, IEEE Comput. Soc. Press.

[67] Ashok K. Goel, Todd J. Callantine, Murali Shankar, and B. Chandrasekaran. Representation, organization, and use of topographic models of physical spaces for route planning. In *Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications: Miami Beach, FL: 24-28 February 1991,* pages 308-314, Los Alamitos, CA, 1991. IEEE, IEEE Comput. Soc. Press.

[68] Hong Li and Marc Dahan. A case-oriented expert system for materials handling. In A. Croisier, M. Israel, and F. Chavand, editors, *1993 CompEuro Proceedings. Computers in Design, Manufacturing and Production: Pris-Evry, France: 24-27 May 1993*, pages 523-530, Los Alamitos, CA, 1993. IEEE Comput. Soc. Press.

[69] Kangjai Lee, Soochan Hwang, Kyhyun Um, and Sungyul Park. An extraction method of analogical information for efficient knowledge processing. In Yuan Baozang, editor, *TENCON '93. Proceedings of the 1993 IEEE Region 10 Conference on 'Computer, Communication, Control and Power Engineering': Beijing, China: 19-21 October 1993,* volume 2, pages 644-647, New York, NY, 1993. IEEE.

[70] Ludmilla Kopeikina, Richard Brandau, and Alan Lemmon. Case based reasoning for continuous control. In Kolodner [80], pages 250-259.

[71] Georgios P. Lekkas, Nicholas M. Avouris, and Loizos G. Viras. Case-based reasoning in environmental monitoring applications. *Applied Artificial Intelligence,* 8(3):359-376, Jul-Sep 1994.

[72] Srinivas Krovvidy and William G. Wee. Wastewater treatment systems from case-based reasoning. *Machine Learning*, 10(3):341-363, March 1993.

[73] Stamos T. Karamouzis and Stefan Feycock. An integration of case-based and model-based reasoning and its application to physical system faults. In F. Belli and F. J. Radermacher, editors, *IEA/AIE 92: Industrial and Engineering Applications of Artificial Intelligence and Expert Systems: Proceedings of the Fifth International Conference: Paderborn, Germany June 9-12 1992*, pages 100-108, Berlin, Germany, 1992. Springer-Verlag.

[74] Zhi-Qiang Liu; Yan, F., "Case-based diagnostic system using fuzzy neural network",Neural Networks, 1995. *Proceedings., IEEE International Conference on*, Volume: 6 , 1995 , Page(s): 3107 - 3112 vol.6

[75] Klaus-Dieter Althoff, Ralph Bergmann, and L. Karl Branting, editors. *ICCBR-99: Case-Based Reasoning Research and Development: Proceedings of the Third International Conference on Case-Based Reasoning: Seeon Monastery, Germany, July 27-30 1999,* volume 1650 of *Lecture Notes in Artificial Intelligence*,

Berlin, 1999. Springer.

[76] American Association for Artificial Intelligence. *AAAI-87: Proceedings of the Sixth National Conference on Artificial Intelligence: Seattle, WA: 1987*, San Mateo, CA, 1987. Morgan Kaufmann.

[77] Association for Computing Machinery. *Proceedings of the Third International Conference on Artificial Intelligence and Law: Oxford, England: 25-28 June 1991*, New York, NY, 1991. ACM Press.

[78] Cognitive Science Society. *Program of the Ninth Annual Conference of the Cognitive Science Society: Seattle, Washington: 1987,* Hillsdale, NJ, 1987. Lawrence Erlbaum Associates.

[79] DARPA. *Proceedings of a DARPA Workshop on Case-Based Reasoning: Pensacola Beach, Florida: 1989,* San Mateo, CA, 1989. Morgan Kaufmann.

[80] Janet Kolodner, editor. *Proceedings of a DARPA Workshop on Case-based Reasoning: Clearwater Beach, Florida: 1988,* San Mateo, CA, 1988. Morgan Kaufmann.