South African Computer Journal
Number 23
July 1999

Suid-Afrikaanse Rekenaar-tydskrif
Nommer 23
Julie 1999

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarvereniging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

World-Wide Web: `http://www.cs.up.ac.za/sacj/`

Subscriptions

	Annual	Single copy
Southern Africa	R80.00	R40.00
Elsewhere	US$40.00	US$20.00

An additional US$15 per year is charged for airmail outside Southern Africa

to be sent to:

Computer Society of South Africa
Box 1714, Halfway House, 1685
Phone: +27 (11) 315-1319 Fax: +27 (11) 315-2276

Guest Editorial
Computer Science and Information Systems: The Future?

Philip Machanick

Department of Computer Science, University of the Witwatersrand, South Africa
philip@cs.wits.ac.za

1 Introduction

As president of the South African Institute for Computer Scientists and Information Technologists (SAICSIT), I have visited a number of campuses and companies, in an attempt at arriving at a general assessment of the state of our subjects in South Africa.

An issue which I consistently pick up is that while everyone seems to think that computer-related skills are extremely important and in short supply, our academic departments are also extremely under-resourced.

At the last Southern African Computer Lecturers Association (SACLA) conference (28-29 June, Golden Gate), I had the opportunity to discuss the problems other academics see. This editorial lists some of the problems reported at SACLA, and proposes a way forward.

2 Problems

At SACLA, I led a discussion of problems seen in our academic departments.

There was wide agreement that both Computer Science (CS) and Information Systems (IS) departments were under pressure to increase student numbers (massification), and were seen as cash cows to prop up less popular subjects. It was broadly agreed that staffing was a critical issue: too few posts for the workload, salaries way out of line with industry (half or less, as compared to the US, where an academic salary may be 80% of an industry salary). Recent graduates often make more than professors which makes it hard to persuade our students to become academics (even to do higher degrees). Attracting a recent PhD with a sense of adventure is may be possible, but attracting experienced people used to earning a salary in a strong currency is hard. IS jobs are worse than CS, as the skills required are more like those in business. Support staff salaries are an even harder issue: their skills relate even more directly to job descriptions in industry.

A problem in addressing our concerns is that we are so overworked that we don't have time for "politics": academics with no students have time on their hands, but we don't. More industry support not only with directly addressing problems but with taking on university administrations would be useful, but they too have major problems and don't have free time.

3 Solutions?

Solutions are harder to identify than problems.

The SACLA session ended with a proposal that we conduct surveys of our institutions and businesses, to find out what the problems are, as a starting point for going to university administrations, government and business.

Another idea was to attempt to find common cause with business in taking on problems they have in common with academia, including the skills shortage, the insufficient capacity of our education system, and dealing with employment equity.

One of our biggest difficulties is to free up time to deal with issues such as resource allocation within our universities. The "competition" is frequently other academics with time on their hands, since they have too few students, and therefore are in a position to spend time looking after their interests.

What is needed now is some thought about how to pull ourselves out of the mess we are in. In particular, we need strategies to exploit our strengths: our high demand among students, the high demand for the skills we produce and the ubiquitous applicability of computer technology.

Given the wide use of computers, it would seem obvious that our areas should be strongly supported by a range of role players, yet the fact that so many different groups are interested in computer technology in one way or another has tended to fragment efforts to enhance our industry and academic institutions.

Clearly, from conversations I have held, some departments are in much better shape than others. Even so, some kind of collective effort is likely to achieve more results than if we allow ourselves to be pushed around as individuals. Addressing the fragmentation of efforts seems a worthy goal in itself, to reduce duplication and contradictory goals.

I appeal to anyone who has constructive ideas on how to take our subjects forward to contact me. Let us work on building ourselves up. The economy depends on us, much more than on most other academic disciplines. It's time we made that point, and made it strongly.

SAICSIT'99

South African Institute of Computer Scientists and Information Technologists

Annual Research Conference 17-19 November 1999

Prepare for the New Millennium

Is there life after y2k?

Mount Amanzi Lodge, Hartebeespoort

near Johannesburg and Pretoria

keynote speaker: Barbara Simons, ACM President

and many other local and international speakers (academic and industry)

Call for Participation

papers in a many areas of Computer Science and Information Systems are expected

Price Waterhouse Coopers prizes: Best Paper R10000 • Best Student Paper R5000

please check the conference web site for accepted papers:

http://www.cs.wits.ac.za/~philip/SAICSIT/SAICSIT-99/

To Register

go to the conference registration web page:

http://www.cs.wits.ac.za/~philip/SAICSIT/SAICSIT-99/reservation.html

or contact **SAICSIT'99 Secretary** *for details:*

Department of Computer Science, Senate House 1137
University of the Witwatersrand
Jorissen Street
Wits, 2050
South Africa

phone (011)716-3309 fax 339-3513 (international: replace 011 by 27-11)

saicsit99-info@cs.wits.ac.za

Dates and Publication Details

early booking deadline: 14 September • on-site registration starts: 17 November 1999

workshops, tutorials 17 November 1999 • paper sessions: 18-19 November 1999

papers will appear in a special issue of South African Computer Journal

sponsors

NRF PRICEWATERHOUSECOOPERS

Research Article

Indexing in a Case-Based Reasoning System for Waste Management

K.L. Wortmann[a], D. Petkov[b] and E. Senior[c]

[a]*School of Mathematics, Statistics, Computer Science and Information Systems, University of Natal, South Africa,*
petkov@comp.unp.ac.za
[b]*International Centre for Waste Technology (Africa), University of Natal, South Africa*

Abstract

This paper presents a summary of research on indexing in Case-Based Reasoning (CBR) applied to the domain of waste management. Indexing theory and techniques are covered in order to position the results. The chosen application domain is discussed very briefly in order to demonstrate the applicability of CBR to it. Indexing techniques implemented are described and discussed. Testing methods and summarised results are discussed with respect to evaluating the success of the indexing techniques, and to identify areas of future research.
Keywords: *Case-Based Reasoning, Indexing, Waste Management*
Computing Review Categories: *I.2, J.2, H.4.2*

1 Introduction

Case-Based Reasoning (CBR) is an Artificial Intelligence (AI) technique which solves a problem by retrieving stored past problems and solutions (cases) which are relevant to the current problem and reusing their solutions to solve the new problem [1, 17]. The basic premise of CBR is that to solve or help solve a new problem, past problems and their solutions which contain information relevant to the new problem are retrieved from storage, and presented for solution construction [4]. To facilitate such reasoning, the basic structure of a CBR system consists of a knowledge base in which past problems and solutions are stored, together with procedures for retrieving relevant problems and solutions [11]. This knowledge base is generally referred to in the literature as Case Memory, while each encoded problem description and solution is referred to as a Case.

Environmental management is an issue of global importance which has proven to be a fruitful application area for computer science and information systems [24]. Nationally, hazardous waste management has been identified as an area in need of particular attention [8]. At national, regional and corporate levels, all aspects of hazardous waste handling are in need of attention. Even where the handling cannot necessarily be viewed as wrong or inadequate, areas which could be improved through the application of information technology can still be found.

This paper presents the results of research on indexing in CBR applied to the field of decision making for pre-transportation handling of hazardous waste. To the best of the knowledge of the authors there are no other reports on applications of CBR to this area. It discusses some technical aspects of indexing that are not readily available in the literature. Two new approaches to indexing are suggested and evaluated. The paper continues with:

- An overview of indexing theory and techniques in order to place the techniques presented here in context;
- A short discussion of the problem of hazardous waste handling in the field in question and how CBR is applicable to solving problems in this area; and
- A description of the design of the prototype system, and the results of its testing and possible implications for future research.

2 Indexing in CBR systems

2.1 Overview

The basis of a CBR system's problem solving capability is the ability to retrieve past situations appropriate to a new situation [4]. The issue of ensuring that the correct cases are retrieved at the correct times is referred to as the indexing problem [15].

The indexing problem can be divided into two areas according to Kolodner [16]. The first is the assignment of labels to cases to ensure that they are retrieved at the appropriate times. The second is the organising of the case library in such a manner that retrieval of cases is done most efficiently (this will be referred to as memory organization). Memory organization also falls under the considerations of retrieval algorithms (and speed of search).

These two indexing issues generally address different goals. The first issue addresses the need to ensure that any one case is available for all areas in which it might teach a lesson. In other words, it tries to ensure that the widest

range of possibilities is covered. This attempt at high coverage can possibly lead to a speed penalty. The second issue, in contrast, attempts to arrange the case library (or rather the indexes) in such a way that retrieval speed is optimized. As with most software design, this speed improvement can lead to a loss in quality, especially as flexibility is lost [4]. It should be noted, however, that organising the case library (eg. into a hierarchy) is not necessarily only due to speed considerations. CHEF [12], for example, used generalisations to facilitate case adaptation. This is an example of memory organisation specifically to assist in indexing rather than to enhance speed of retrieval.

In the context of the two major issues in indexing, the following areas can be discussed:

The desirable qualities in indexes - the concepts which should be addressed and the necessary information which should be captured when the indexing method for a reasoner is decided on, and when indexes are assigned for a specific case.

Choosing an indexing vocabulary - Once it has been decided what knowledge needs to be captured by indexes, it must be decided what vocabulary should actually be used for indexing.

Methods for index assignment - This is the question of which technique(s) should be used when actually assigning indexes to a case.

Memory organisation strategies - How cases and memory should be organised.

It can be seen that there are a wide number of issues to consider when designing indexing and memory organization. In addition, literature indicates that solutions are largely domain specific. For instance, indexing could be a simple matter of assigning weights to fields in a flat memory structure, as can be seen in [22], or a hierarchical memory organization such as the second implementation of Battle Planner [10], or multiple indexes for cases depending on the current goal, as in OCCAM [20, 3]. Due to space constraints, this paper will concentrate only on methods for assigning indexes.

2.2 Indexing Methods

Views on actual methods for assigning indexes to cases tend to vary widely in the literature. Kolodner, for example, split indexing methods into two main areas, namely indexing by hand, and indexing by machine, which was then further divided into indexing using checklists, difference based indexing, explanation based indexing, or a combination of the techniques [16]. Barletta, in contrast, differentiated further between cases according to the available knowledge and type of reasoning goal, and classified techniques into nearest neighbour, inductive and knowledge guided approaches [4]. Hansen, Meservy and Wood preferred to split indexing into surface feature indexing and structural indexing[13].

These apparently different ways of classifying, essentially, the same issue are indicative of the great diversity of views in CBR. However, examination of the classifications shows that there is a high degree of overlap between the various classifications, which in some cases, refer to the same concepts. For the purposes of this communication, the Barletta [4] method will be followed but in a modified form by the integration of it with the classifications of Kolodner [16] and Hansen, Meservy and Wood [13].

2.2.1 Fixed Indexing Techniques

Nearest Neighbour Approaches (Fixed Matching)

Nearest neighbour approaches in this context refer to indexing used when there is traversal of a flat case memory (eg. a single data base table), with the problem case matched to each case in case memory. This method is described for example in [13]. Cases retrieved are those which most closely match the current problem. Here, indexing is performed by deciding on the generally important features of all cases as a group, and then assigning weights to these fields. Note here that cases are not weighted according to individual features but only at a global level.

There is a major problem with this approach as discussed in [28]. In many domains, the importance of various features varies from case to case. Thus, each case should, ideally, have its own set of weights which identify its discriminating features. One such a problem was encountered in Battle Planner [10] when domain experts found it extremely difficult to assign weights to fields when a nearest neighbour algorithm was used. As weights were modified to suit one set of cases, so they failed to work effectively on another. The technique was therefore abandoned due to the failure to find a successful consistent weighting scheme. This problem was solved by an inductive approach which will be discussed below.

Using Checklists

Checklist-based indexing [16] bears little resemblance to a fixed-weight approach to indexing such as nearest neighbour search. There is, however, one feature which is shared between the two methods, and this will be discussed briefly.

In the checklist approach, at build time a list of features is created on which every case is indexed. Thus, when a new case is indexed, the checklist is referred to and those fields specified in the checklist are the fields on which the case is indexed. In the case of CHEF [12], this means that all cases are indexed on ingredients, method of preparation, as well as other fields. Thus every case is indexed on these same fields, namely those identified in the checklist.

It is this fact that in all cases the same fields are indexed that provides common ground with the fixed-weighting nearest neighbour approach. In fixed-weighting, all cases have the same global weighting thus, effectively,

all cases are indexed on the same features. This leads to problems of determining good weights. In a checklist approach, the same type of technique is used in so far as it is presumed that the same fields in all cases are teaching the lessons required.

Checklist indexing is not, however, nearly as restrictive as fixed-weighting nearest neighbour indexing. In the latter, the memory structure is flat, and cases are indexed for one goal only. Checklists do not restrict in this manner. For example, there is no reason why a case should only be teaching a single lesson (and thus being indexed once). Multiple goals (therefore multiple indexing) can be achieved by simply creating multiple checklists where each checklist is designed around a different context. In this way, any case can be retrieved for a number of different goals.

In addition, from a memory structure point of view, checklist indexing does not prevent the structuring of memory in a hierarchical manner. A good example of this is CHEF [12], where indexes are generalised (eg. beef becomes meat), and memory becomes hierarchical in nature.

Checklist indexing thus has in common with nearest neighbour fixed weighting the concept of fixed feature matching. However, unlike nearest neighbour, the technique is not restrictive in terms of memory structure nor does it restrict the number of contexts in which a case may be used to teach a lesson.

2.2.2 Inductive Approaches

Inductive indexing, also called structural indexing [13] is based on the use of an algorithm to index cases and arrange memory based on differences between cases. ARCHIE [21] uses nearest neighbour and inductive approaches. Battle Planner's [10] problems of assigning weights to fields were solved by using inductive indexing. In this approach, indexing is generally performed by providing a representative set of cases to an inductive algorithm [13]. This algorithm then analyses the cases and, based on the decision required from the cases, determines which features best discriminate between cases and creates a decision tree based on these features.

This technique provides a number of advantages, as discussed by Barletta [4], and is seen in the development of Battle Planner [10]. The first is that cases are automatically analysed for their predictive features. The difficulty of identifying the important features by hand, as was initially done in Battle Planner, are avoided. The second is that memory can be organised hierarchically. This greatly reduces the retrieval time of cases which was, initially, also a problem with Battle Planner.

It is intuitive that inductive approaches provide advantages over the more simple nearest neighbour weighting approaches as there is a progression from a static indexing method to one generated from the specific requirements of the system. Unfortunately this advancement also brings its disadvantages. Firstly, the goal or outcome must be well-defined [4]. Secondly, there must be enough cases to give adequate coverage of each type of goal otherwise comparisons will not be made effectively [4]. A third drawback, identified by Barletta [4], is that the inductive analysis can take a lot of time.

Difference-based indexing [16] appears to refer to the same basic concept as inductive indexing, namely that indexing is performed by comparing cases. The basis of this theory is that the purpose of indexing is to differentiate between cases (in a manner which supports the reasoning goal(s)). It, thus, makes sense that a possible technique for indexing is to index on the differences which a case exhibits, eg. if a case's value for a particular field is the normal value, do not index on it. However, if this value differs from the norm, it differentiates the case from the norm and, thus, should be indexed on this feature.

2.2.3 Knowledge-Based Indexing

Knowledge-based indexing (featuring as explanation-based indexing in [16]) is another way of overcoming the problems of a fixed weighting nearest neighbour indexing approach, and inductive/difference-based techniques. In all techniques discussed thus far, there is no analysis of individual cases to discover how each is predictive. Even though inductive indexing techniques decide which features of a case are predictive based on initial individual comparison, they do so by making use of a model of features which are usually predictive [16] and do not analyse each specific case to determine which features are predictive for that specific case. This means that features which are generally predictive will be indexed on even in cases where they are not and that features which are generally not predictive will not be indexed on even in cases where they are predictive [16]. This leads to the situation where inappropriate cases are retrieved or where appropriate cases are overlooked. Such results do not necessarily lead to failure but rather to sub- optimal performance.

The premise of knowledge-based or explanation-based indexing is that each case is analysed individually to determine which features of that case are predictive, and the case is then indexed on these features. This is achieved by building sufficient explanatory information into the reasoner [4].

In this communication, an alternative name was suggested for the nearest neighbour indexing approach, namely the fixed weighting approach. This is because Barletta's [4] description of a nearest neighbour approach indicated that each case is indexed via fixed weighting, thus each case is identically indexed. It is contended, however, that a nearest neighbour matching system should not have to be approached by using a fixed weight scheme as described by Barletta [4]. Instead, a knowledge guided method could be employed whereby each case is assigned its own weighting. This method was in fact suggested by Barletta [4] as a desired approach to the nearest neighbour method. However, the link is not explicitly made to knowledge guided indexing methods. It is, thus, contended here that assigning individual case weights could be achieved

in a nearest neighbour system by making use of domain knowledge (if available). Thus, it is possibly misleading to refer to the first method as a nearest neighbour approach, therefore, fixed weighting approach is suggested as an alternative.

There are certain problems to knowledge guided techniques. The first is that the explanatory knowledge needed must be available and representable [4]. The second is that it is often difficult to encode enough of this knowledge to effectively index all possible cases [4].

A solution was, however, presented in Kolodner's first indexing technique - indexing by hand, without it being explicitly presented as such [16]. This technique is, in fact, a knowledge guided technique; simply it is not an automated one. Instead of the reasoner having explanatory knowledge built into it which will be used to analyse a case, it is left to the user to identify for the reasoner the indexes for a case. The user, therefore, is required to follow the guidelines for desirable qualities of reasoners and, thus, must identify the predictive features of the input case (for all possible reasoning tasks), generalise where possible (only if the reasoner is using generalisation, which will be discussed below), and then translate these features into the reasoner's indexing vocabulary. While this process might appear tedious, and suffers from the fact the user involvement is required in the system learning process, it is felt that it is highly applicable to reasoners which provide decision support as proposed in [15].

If we consider this simplified process, we can see that it is in fact a knowledge (explanation) based approach, with explanations provided by the user, not the reasoner. This of course overcomes the issues of representing and encoding sufficient explanatory knowledge.

The above issues were therefore taken into consideration in the design of the indexing techniques which are presented in this paper.

3 Review of the current decision making process for pretransportation handling of hazardous waste and justification for the use of CBR

3.1 How decisions are made in the application domain

The research was carried out within the framework of procedures for pretransportation handling of hazardous waste, observed at a large waste technology company. The current decision making process can be described as follows:

When a batch of waste is received from a client, a data sheet is used to record information related to the batch. Firstly, the basic client information (name, address, date, etc) and initial waste description are recorded on the data sheet. Once this has been done, a sample of the waste is passed on to the laboratory for analysis. Having captured all the data about the batch, including the outcomes of the analysis, the data is handed over to an expert. It is now her/his job to provide solutions for the case. These solutions relate to issues such as disposal instructions, mode of transportation, type of container to be used, and the information to be displayed on the transportation vehicle. The solutions may have considerable economic and environmental impact. This justifies the need for seeking their improvement through advanced technologies.

The first feature to note about the decision making process in place is that it is totally uncomputerised. While it can not be concluded that because a process is uncomputerised it can be considered problematic or flawed, examination of the current procedure does reveal a number of possible problem areas or areas for improvement which computerisation could aid. These are all related to the point that access to old solutions is a tedious process as it involves retrieval of paper sheets from files, and an expert must remember the existence of a past problem solved.

Essentially, three issues can be highlighted:

- Due to the simple filing of data sheets, old solutions are not used actively to solve new problems. Instead, the expertise and memory of the expert handling the current problem is relied upon. As such, past experience is not readily available for current decision making;
- Problems which have occurred in the past in solutions provided are not actively identified again in new situations. Again, the expert is relied upon to identify these problem areas. Thus, failures which have occurred before are not always remembered; and
- When a new person takes over the role of producing solutions, validation of the solutions suggested is not easily achieved unless another expert is available. This is because the new people being trained in decision making do not have easy access to past experience.

These three issues highlight a number of important points. If, for instance, an expert has solved a problem before, and then encounters the problem again, there could be a number of scenarios. Firstly, the expert may not remember having handled the problem before and thus provides an independent new solution, which may lead to subjective errors. Alternatively, the expert may remember having solved the problem before and must now remember where this past solution is filed. Another issue is encountered when the same problem is handled by different experts and results in different solutions. Without access to the other solutions, an expert cannot easily verify her/his solution.

3.2 Suitability of CBR to the application domain of hazardous waste handling

Examination of the manual process and problems described in the preceding subsection illustrates two points. Most problems/areas for improvement result from inefficient use of existing expertise. Also the specific applica-

tion domain is rich in past expertise which is recorded in the form of data sheets. Therefore, CBR is applicable to the domain. Further reasons are listed below.

Firstly, information in the application domain is not completely defined. Mixed wastes are quite often dealt with. As there are, potentially, endless combinations of wastes, any decision support system applied to the domain should cater for possible new combinations. For a CBR system, this is handled simply by being able to learn new problems and their solutions as they occur. This advantage includes the learning of any possible failures. Secondly, CBR is said to model the human reasoning process and is, thus, easier for a user to employ. A third issue is justifications for decisions. As discussed above, by providing access to actual past solutions, CBR facilitates justification for the solutions produced. These considerations all led to the conclusion that the application domain chosen was valid for the testing of CBR indexing techniques.

4 Description of Implemented Case Representation and Indexing Techniques

For the purposes of this research, a single case representation was used which represented the case knowledge completely, and allowed for the implementation of indexing techniques, the primary aim of the research. Five indexing techniques were implemented and tested in two prototype systems. One of them was using a commercially available CBR engine, ESTEEM. The latter is a good value for the money according to [27]. The second was developed by the first of the authors.

4.1 Case Representation

It was decided, after consultation with the company used as the test domain, that the data sheet as currently used by them would be employed in as unmodified a form as possible for case representation as, firstly, the format already lent itself well to a database format (a valid case format) and, secondly, with a view to real-world application, it was decided that unnecessary deviations from the existing data layout would be counter-productive. As a result, construction of a case was largely a matter of creating a field in the case for every field in the data sheet. In addition, the case was separated logically into a REPRESENTATION and SOLUTION section, each consisting of a number of fields. There were, however, some modifications necessary to this data sheet. Some fields from the original sheets were omitted as they did not hold information which was used in solution derivation. A case description contained 40 REPRESENTATION fields. Their values represent the constraints on the goal of the waste disposal.

In addition, certain fields were added to the original uncomputerised case representation. All these fields are SOLUTION rather than REPRESENTATION fields. These fields are used to store the reasons for the solutions stored in the SOLUTION fields and are used for certain indexing techniques and deriving solutions to new problems. Specifically, for each SOLUTION field there is a corresponding NEW field which notes which REPRESENTATION fields were responsible for that particular solution being chosen.

4.2 Indexing Techniques

As stated, a case in the CBR system consists of two sections - REPRESENTATION and SOLUTION. The fields in the REPRESENTATION section determine the solutions in the SOLUTION section. Since the goal of indexing (in general) is to identify the predictive fields in a case, the goal of indexing in this case base is to identify the REPRESENTATION fields which are predictive of the solutions in the SOLUTION fields of the cases. The five techniques discussed here represent somewhat of an evolution in approach. The first three of the techniques, described briefly below, are examples of single global indexing, while the last two are examples of single case-specific indexing and multiple case-specific indexing.

Equal Weight Nearest Neighbour (EWNN)

This is the most basic technique for indexing in CBR. For this technique, all fields in the REPRESENTATION section of a case were assigned a weight of one. In other words, no field was given any more importance than another field. This technique is equivalent to the SIM-EVEN technique described in[22], and shall be referred to as EWNN in the custom implementation and EWNN-EST for the ESTEEM implementation.

Fixed Weight Nearest Neighbour Using Expert Judgement (FWNN-EXP)

This technique is equivalent to the SIM-F technique in [22] or the weighting method initially attempted in [10] and is also a standard method for the nearest neighbour technique. For the provision of the necessary weights for this technique, a person currently involved in the decision making process at the company concerned was approached. This person was asked to assign a weight to each field in the REPRESENTATION section of a case based on its importance relative to all other fields in the REPRESENTATION section. The more important the field, the higher the weight.

Fixed Weight Nearest Neighbour Using the 80-20 Rule (FWNN-80:20)

This technique offers an interesting comparison to the FWNN-EXP technique. Unlike in FWNN- EXP where all fields were weighted by an expert, it was decided here to make use of the 80-20 heuristic rule for weighting of fields. In this innovative approach the authors have used a common heuristic in Operational Research meaning that

usually 20% of the data contribute to up to 80% of the information. Thus, for this technique, weights were assigned only to eight of the forty problem representation fields.All other fields were assigned a weight of zero. This technique is referred to as FWNN-80:20 for the custom developed system and FWNN-80:20-EST for the ESTEEM control prototype. These fields were chosen by the authors firstly by taking into account advice from the domain expert, and secondly by examining all the data contained in the data sheets used for case base construction. As such, the fields chosen were influenced heavily by the contents of the case base used. This fact may be viewed as a drawback or as a positive feature depending on the circumstances as will be seen in the section on the discussion of the results.

The three above techniques represent each a global weighting scheme. In other words, having examined the case base as a whole, weights are assigned to the various fields once. Each case then gets assigned the same weight for a particular field. The following techniques used cases which are weighted individually.

Variable Weight Nearest Neighbour (VWNN)

The fields added to the original case descriptions during the design of the CBR system hold the information which indicates which fields in the REPRESENTATION section were responsible (according to the domain expert) in that particular case for the solution. The contents of these fields are in fact a ready-made means for indexing a case. This is used in the Variable Weight Nearest Neighbour (VWNN) technique for indexing. Instead of using a global weighting of fields in the REPRESENTATION for indexing, it uses the values of the NEW (i.e. reason) fields in the SOLUTION section to determine the indexes for a case.

Effectively, what happens for any case is as follows. Firstly, all REPRESENTATION fields get an initial weighting of zero. To then assign weights to fields, the following process is followed. Each time a REPRESENTATION field and its value are listed in a NEW (reason) field, the weight of that REPRESENTATION field for the case is increased by one. Thus, to index the entire case, each NEW (reason) field is scanned to obtain all REPRESENTATION fields listed, along with the number of times they are listed. This information then weights only the predictive REPRESENTATION fields for that case, the weight determined by the number of occurrences of that REPRESENTATION field in the SOLUTION NEW (reason) fields

This approach has a resemblance to knowledge based indexing techniques discussed earlier. It is justified by the fact that the predictive features vary from case to case. Using the VWNN method, each case is weighted (and thus indexed) individually, and only on the predictive features for that case. In other words, VWNN is more flexible.

Separate Field Variable Weight (SFVW)

The final technique discussed here is an extension of the previous one and is a new approach introduced by the authors. As with VWNN, Separate Field Variable Weight (SFVW) also makes use of the NEW (reason) fields contents in a case. As such, the principle is the same. The essence of the difference (or extension) is described below:

In VWNN, a single case-specific index is created for each case by using the NEW (reason) field contents (as compared to the first two techniques which use a single, global index). SFVW extends this principle by using these NEW (reason) fields to create multiple case-specific indexes. This is achieved fairly simply in the following manner.

In VWNN, all NEW (reason) fields are scanned once and their contents used to weight certain fields in the REPRESENTATION section, thus creating a case-specific (but single) index for the case. In SFVW each NEW (reason) field is used separately for case weighting. In other words, we take the first NEW (reason) field, examine its contents alone, and then weight the case in the same manner as VWNN. We now have the case weighted for the SOLUTION field which the NEW (reason) field contained reasons for. By weighting separately for each NEW (reason) field, we effectively create a case-specific index for each of the SOLUTION fields of the case. Thus, we can say that SFVW creates multiple, case-specific indexes.

The five techniques described above were implemented in a custom built system using Borland Delphi 1.0. These were all incorporated into a single program called HACA - The Hazardous Chemicals Advisor. Cases for HACA are stored in a single Paradox table, one record per case, and all coding was done in Object Pascal, Delphi's coding language. The purpose of these custom implementations was to assess the performance of the five techniques with reference to each other.

As a way of validating some of the capabilities of HACA, two of the techniques, EWNN and FWNN-80:20 were implemented using the ESTEEM, v. 1.1 CBR shell as mentioned before. The last two of the five techniques described above were not supported in this version of ESTEEM. The remaining FWNN technique is very similar to the FWNN-80:20 approach and thus there was no need for its implementation in ESTEEM. All experiments were executed under Windows 95, running on a 486 DX4/100 PCI IBM PC compatible with 48MB of RAM. The results of the experiments are presented in an abbreviated form in the next section.

5 Experiments and Results

5.1 Formulation of Experiments

O'Leary [18] reported that validation of a case-based system generally involves comparing system outputs to human experts or machine solutions. The first approach was facilitated through the availability of past data and the co-

operation of experts from the company involved. As actual data sheets are used for case representation, access to human expertise is readily available. The second approach was followed through the implementation of those techniques, for which it was possible, by using a commercially available CBR shell.

Experiments were designed with three goals in mind, namely:

- Comparison of some of HACA's indexing techniques with similar possible techniques using ESTEEM for the purposes of validating the corresponding HACA components.
- Comparison of the five indexing techniques as implemented in HACA to determine which of the proposed indexing techniques performed best; and
- Decide what implications the results have for future research.

The ESTEEM implementations of the EWNN and FWNN-80:20 indexing approaches used the same case base and test cases as HACA. They were only used as controls for the corresponding techniques, implemented in the custom built system. The assumption was made that as a commercial CBR tool, ESTEEM's indexing and retrieval would function correctly. As such, the ESTEEM implementations were used to provide figures against which the corresponding techniques from the custom implementations could be compared. Thus at least a partial validation of HACA against a commercially available system was achieved.

A testing approach similar to that in [10, 7, 22] was employed to evaluate the techniques. As HACA's intended role is as a decision support tool (following [15]), and not an expert system, no attempt is made to build solutions automatically. The testing involved using a randomly selected subset of actual cases stored in the case base as test inputs to the system, determining solutions based on retrieved cases and then gathering results deemed relevant.

A case base of ninety cases was constructed for testing purposes. As our results showed later, this case base was sufficient for testing the indexing techniques under concern. On the other hand, for industrial type implementations a larger case base is preferable. There are usually practical limitations to this. In our case they were related to the need to use an expert to provide the content of the solutions fields which were not included in the original data sheets used in the manual operation. Similar or smaller case base sizes were used in [6, 14, 21, 26]. This however is only a rough indicator and cannot be used for a rigorous validation of the size of the case base as each application domain is of different complexity and may require different size of the case base.

The literature on CBR does not indicate any strict rules for the size of the sample that should be used for testing purposes. Thus in [7, 10, 22] just 10% of the original cases in the case base were used for testing. On the other hand, in [26] a subset of 136 cases was used for testing, which represents 96% of the entire case base. In our experiments was chosen a compromise value of 22% as a test set of 20 cases was chosen at random from the case base of 90 cases. This same set was used as an input for tests on all techniques implemented in ESTEEM and HACA.

Three characteristics were gathered during the experiments: retrieval time, case similarity and prediction accuracy. The first characteristics measured was *Retrieval time*. These data were used to compare the techniques implemented both in HACA and ESTEEM and to determine whether any of the techniques as implemented in HACA effected a degradation in retrieval speed compared to the others.

The computation of *similarity (matching)* between an input and stored case in the systems is done by calculating matchings between relevant fields, and then computing a matching for the overall stored case. The Symbol technique was used for *fields similarity* [22], which is simply expressed as:

$$similarity(f_{input}, f_{stored}) = \begin{cases} 1 & \text{if } f_{input} = f_{stored} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In other words, two fields either match exactly or do not match.

Fields which are empty (i.e. either empty or have the value 'No Test') are handled differently here from [22] however. This is because fields with no value are not predictive of a solution. As such, empty values should not contribute to matching at all. Thus, when two fields which are empty are matched, they are not counted. As such, we have:

$$similarity([empty/No\ \ test],[empty/No\ \ test]) = No\ \ contribution \quad (2)$$

However, if one field has a value and the other is empty, then similarity is considered to be zero:

$$similarity([empty, No\ \ test], any\ \ non-empty\ \ field) = 0 \quad (3)$$

To determine overall case matching, the weighted normalised sum, as suggested in [22], was used in the system, converted to a percentage. The formula is as follows:

$$casesimilarity(c_a, c_b) = \frac{\sum_{f \in F} w_f \times similarity(v_{fa}, v_{fb})}{\sum_{f \in F} w_f} \times 100 \quad (4)$$

where c_a and c_b are the cases being compared, F is the set of fields being compared, v_{fa} is the value of field f in case a, v_{fb} is the value of field f in case b, and w_f is the weight of field f.

It was decided that *prediction accuracy*, as used in [10] and [22] (the latter authors used the reciprocal of this and called it error rate), would be used to assess performance. As there are a number of solution fields in a case and, possibly, a number of components to each solution, it is possible to make some correct and some incorrect predictions

for the same case. Thus, *prediction accuracy* is calculated as:

$$\frac{N_S - N_G}{N_S} \quad (5)$$

where N_S and N_G are respectively the total number of solutions and the total number of guesses; Total number of solutions is: Maximum of the total number of solution components of the input case and currently matched stored case. For SFVW, this total refers to a field while for all other techniques this refers to all SOLUTION fields. Examples of calculations can be found in [29].

The actual experiments were organised according to the following plan:

- Twenty cases were chosen at random from the case base. This same set was used for tests of the relevant techniques, implemented in HACA and ESTEEM;
- Each case was used as an input to the system but, unlike other systems, this case was not removed from the case base. The reasoning for this was that for retrieval the objective was to examine whether an exact match present in the case base would be retrieved accurately. Obviously, this ideal case was not used for solution construction;
- Retrieval of each case from the full case base using each technique was attempted on a current input case. The threshold matching level for a success was 0.3 (30%);
- For the first four techniques (Equal Weight Nearest Neighbour (EWNN), Fixed Weight Nearest Neighbour using Expert Judgement (FWNN-EXP), Fixed weight Nearest Neighbour using the 80-20 rule (FWNN-80:20) and Variable Weight Nearest Neighbour (VWNN)), the top ten cases of those retrieved for each of the 20 test cases (or all cases when 10 or less were retrieved) were taken and used to construct a solution for the input case;
- For Separate Field Variable Weight (SFVW), the top three cases for each SOLUTION field were used to construct a solution for that field due to the restrictions on the time necessary for the involvement of the domain expert in constructing a solution; and
- The following data were then calculated for the first four techniques, including those implemented in ESTEEM and HACA :
  - The time taken for case retrieval for each of the 20 input cases and an indexing technique. The average times for all cases are shown in table 1, result RT.
  - The matching of all the cases retrieved above the threshold for each input case and indexing technique as a percentage. The average matchings for all cases are in table 1, result AM. One may note that here and for the rest of the rows of table 1 there are no values for the EWNN-EST implementation as no cases were detected in its testing for which was achieved the threshhold value of 30.
  - The matching of the top ten (or less) cases retrieved for each input case (i.e. the cases used for solution construction). The average matchings for all cases are shown in table 2, result AMT.
  - The number of cases retrieved above the similarity threshhold value of 30 for all input cases and techniques. The average for all cases is in table 1, result NCR.
  - The prediction accuracy for each input case and all techniques. The average results are in table 1, result AS.
- For SFVW, due to its different retrieval nature, the following results were gathered:
  - The time taken for case retrieval. These results are shown in table 1, the relevant result is RT in column SFVW.
  - The average matching of all the cases retrieved above the threshold value of 30 for each separate SOLUTION field. The averaged data are shown in table 2 as result AMSF.
  - The average matching of the top 3 (or less) cases retrieved for each separate SOLUTION field. The average data are shown in table 2 as result AMTSF.
  - The number of cases retrieved for each SOLUTION field. The average data are shown in table 2 as result NCRSF.
  - The prediction accuracy of each retrieval for:
    * Each SOLUTION field in each case. See table 2, result ASSF.
    * Each complete case. See table 1, result AS, column SFVW.

  Please note that table 1 and table 2 contain only the summary data from the experiments. Detailed results for each input case can be found in [29].
- To evaluate performance, prediction accuracies of the techniques were compared using paired t-tests as described in [23].
  - In order to determine whether the custom implemented techniques (i.e. the HACA implementation) were operating as they should, FWNN-80:20 and FWNN-80:20- EST prediction accuracies were compared with a null hypothesis of equality of the mean prediction accuracies (EWNN and EWNN-EST prediction accuracies could not be compared as no cases in the ESTEEM implementation could reach the 30%

Result Code, Dimenson	EWNN-EST	FWNN-80:20-EST	EWNN	FWNN-EXP	FWNN-80:20	VWNN	SFVW
RT (s)	16.7	6.3	3.2	3.2	3.2	3.2	3.2
AM (%)	-	39.5	41.4	42.8	45.5	59.0	-
AM T (%)	-	51.0	53.5	57.5	63.3	85.3	-
NCR (-)	-	44.8	47.5	51.1	55.5	46.6	-
AS (-)	-	0.86	0.78	0.82	0.85	0.87	0.89

Table 1: Retrieval time (RT), average matchings for all cases (AM), average matchings for the top 10 cases retrieved (AMT), number of cases retrieved above threshhold (NCR) and accuracy of solutions (AS) produced as an average for all input cases and all indexing techniques implemented in ESTEEM and HACA as described in each column

Result Code/dimension	**Disposal**	**Site**	**Hazchem**	**Trem**	**Warning**	**Clothing**
AMSF(%)	59.1	0.0	84.2	62.8	64.4	20.0
AMTSF(%)	96.7	0.0	98.5	96.1	60.3	20.0
NCRSF(-)	32.7	0.0	36.9	47.9	2.0	0.3
ASSF(-)	0.93	1	0.75	0.75	0.98	0.98

Table 2: Average matchings of all cases retrieved above threshold (AMSF), average matchings of top three cases retrieved above threshhold (AMTSF), number of cases retrieved above threshhold (NCRSF), accuracy of solution produced (ASSF) for each separate SOLUTION fields as named in each column, for the Single Filed Variable Weight (SFVW) indexing technique

threshold value for a successful retrieval as already mentioned). Since the test sample is of 20 cases, the degrees of freedom $df = n - 1 = 19$. The critical value corresponding to the upper 2.5% region and the lower 2.5% region of the t-distribution with 19 degrees of freedom is found from a corresponding statistical table to be 2.093. The value of the t-test statistic is 0.622. Since it is smaller than the critical value, the null hypothesis is not rejected. Hence the HACA implementation of this indexing technique is validated against the ESTEEM implementation.

- In order to determine the relative performance of the five techniques implemented in HACA, each technique's prediction accuracy was compared to the rest. Comparisons were performed using a null hypothesis of inequality in the means of the prediction accuracies to determine whether the technique with the higher average prediction accuracy could be considered superior to that with the lower prediction accuracy. The critical value corresponding to the upper 5% region of the t-distribution with 19 degrees of freedom is found from a corresponding statistical table to be 1.729. The null hypothesis is that the difference between the mean prediction accuracies of any pair of indexing techniques implemented in HACA is smaller than zero. This means that the second technique has achieved bet= ter prediction accuracy. The opposite hypothesis is that the above difference is greater than zero. The values of the t-test statistics for the comparisons between the different indexing techniques implemented in HACA are shown in table 3. Where ever such a value is greater than the critical value of 1.729, the null hypothesis is rejected. Thus the decision maker can be 95% certain that there is an improvement in the prediction accuracy of the second technique in a given comparison.

5.2 Discussion of the results

5.2.1 Comparisons between ESTEEM and HACA

The purpose of the ESTEEM implementations was to validate the HACA techniques. The comparison between HACA and ESTEEM could be done with the FWNN-80:20 only as EWNN- EST did not produce results above the 30% threshold value for case similarity.

The result of the paired t-test comparing the prediction accuracies between FWNN-80:20 EST and FWNN-80:20 as shown in the previous subsection leads to the conclusion that FWNN-80:20 in the HACA implementation was performing as it should from an indexing perspective. This permitted indirectly the further comparison of the five HACA indexing techniques.

The lower matching and number of cases retrieved in ESTEEM (see table 1) may be due to the difference in the way in which ESTEEM handles blank entries to HACA's handling of NO TEST/blank cells. HACA ignores these fields, rather than letting them detrimentally affect the matching figure.

Retrieval time (table 1) shows results which are encouraging for HACA, but it also shows indications for some possible improvements. In all techniques tested in HACA, retrieval time was the same on average. However,

Technique	EWNN	FWNN-EXP	FWNN-80:20	VWNN	SFVW
EWNN		1.963	2.300	2.754	2.846
FWNN-EXP			1.700	2.120	2.001
FWNN-80:20				1.555	1.345
VWNN					0.586

Table 3: Results of t-test comparisons between the prediction accuracies for five HACA techniques

in ESTEEM, retrieval time dropped from EWNN-EST to EWNN-80:20-EST by 62% (rounded off). There is, thus, the need to improve the retrieval time of HACA.

5.2.2 Discussion of the Five Techniques as Implemented in HACA

Prediction Accuracy

As discussed, prediction accuracy was used as the test of effectiveness of the techniques, with other data providing backup and/or insight into future research issues. An examination of the accuracy of solutions produced from the five techniques (see the last row in table 1) indicates an increase in average values from EWNN to FWNN-EXP to FWNN-80:20 to VWNN to SFVW .

The first comparison to make is between the two case-specific techniques, VWNN and SFVW. It was described how cases have a number of solutions rather than a single one, and that SFVW retrieves for each solution separately. Examination of the cases indicated that reasons for different solutions varied within a single case, so it could be expected that SFVW would perform better. Statistical comparison using paired t-tests (as described in [23]), however, indicated that SFVW was not significantly better than VWNN. One possible reason for this result is simply that SFVW is not a suitable technique for this application domain. There are, however, other issues which relate to the concept of validated retrieval (as described in [25]).

One of them is that in SFVW often many cases which taught exactly the same lesson were retrieved. A possible solution to the improvement of SFVW results would be to implement some sort of validated retrieval mechanism.For instance, cases could be grouped so that those which indicated the same solution for the same reason(s), and then only one candidate case from each of these groupings could be presented. Another issue is that the case base was found to be inconsistent. For example, one case indicated that a TREM card (one of the SOLUTION fields) of 3.15 should be used while another indicated a value of 3.27. Such situations could have led to further degradation in performance of SFVW due to the small number of cases utilized for solution construction. Despite the possible reasons for SFVW not out-performing VWNN, it must be concluded based on the results obtained that SFVW as it stands cannot be considered superior to VWNN.

The next comparison to make was between the case-specific indexing techniques (SFVW and VWNN) and the global techniques (FWNN-EXP and FWNN-80:20). Since it can be noted that the important knowledge is not located in the same fields in each case, it is expected that case- specific techniques would out-perform those using a global technique. Examining table 3 shows mixed results. According to it, for the conventional global weighting mechanism (FWNN-EXP), both case-specific techniques can be seen to be providing better results. This, taken in isolation may lead to a conclusion that case-specific indexing outperforms global indexing.

Comparisons with FWNN-80:20, however, show a different result. From table 3, it cannot be concluded that either of the case-specific indexing techniques out-performs FWNN-80:20. Considering the fact that FWNN-EXP and FWNN-80:20 are very similar in concept, this result is surprising. However, it is felt that there is a valid reason for it. It was pointed out earlier that weights in FWNN-80:20 were chosen based on expert opinion and the specific case base of 90 cases used. Weights in FWNN-EXP were on the other hand chosen based only on expert opinion, without reference to the specific case base used. As such, the implementation in this research of FWNN-80:20 can be viewed as a biassed approach, as it is tailored specifically to the case base used, and not to a case base of any size or diversity. It would be expected that this bias would result in degradation of performance of FWNN-80:20 if the case base was increased in size and diversity, while the other indexing techniques, which were chosen without bias towards the specific case base in use, would not experience degradation.

It is thus felt that, for a flexible expanding case base, it can be concluded that the case-specific indexes do indicate better performance than global indexing, as is illustrated by the significantly higher performance of SFVW and VWNN over FWNN-EXP, the conventional global indexing technique.

On the other hand when there is less likelyhood for inclusion of additional new cases in the case base, a good knowledge of the data in the case base is a precondition for improved performance in terms of retrieval time and prediction accuracy.

Error rates in HACA are not directly comparable with those of other reasoners as they may be implemented for a different application domains which may vary in complexity. However the error rates in other reasoners published in [22, 10, 7] do not indicate any significant differences from the prediction accuracy that is achieved in HACA. The latter can be seen as an indirect rough indicator for the quality of HACA.

Case Matching and Number of Cases Retrieved

Matching figures in table 1 show that matching increased from EWNN to FWNN-EXP to FWNN-80:20 to VWNN. The question is whether this indicates a greater certainty in the usefulness of cases retrieved. From the prediction accuracies (table 1) it can be seen that for the increase in matching figures, there was a corresponding increase in accuracy (although not always a significant increase). For FWNN-EXP and FWNN-80:20, the pattern is not clear. For VWNN the number of cases retrieved (table 1) indicates clearly that while matching figures and prediction accuracy are higher, the number of cases retrieved does not rise correspondingly (it is actually slightly lower). This leads to the conclusion that for VWNN, the higher matching figures are indeed due to a higher certainty in the usefulness of cases retrieved, rather than simply differing matching methods.

For SFVW, a slightly different situation existed. Specifically, as retrieval is done on a by-field rather than by-case basis, a matching figure for whole cases could not be calculated, rather only for parts of cases. These results are shown in table 2. Obviously, these cannot be compared to those in table 1. Nevertheless, there are two issues to point out. The first is that average matching for the top 3 cases (table 2) was higher than for all cases retrieved above threshold (table 2). This would indicate that 3 cases is a possible cutoff in terms of case selection for decision making.

The second point to explain is the low matching figures for the Site and Clothing fields, as well as low retrieval for Site, Warning and Clothing fields (table 2). This was not due to there being no solutions present for these fields in case memory. Rather, it was due to the way certain fields are handled in matching. Specifically, in certain situations, the content of a case does not warrant any special action. In such a situation, a default or STANDARD action is taken. An example would be "Gloves, Boots and Goggles" for protective clothing. In such a situation, STANDARD is entered in the reason field, in this case the Warning field. Obviously, STANDARD will not match any REPRESENTATION fields contents, thus for this field, a matching of zero will occur, and hence cases are not retrieved. Matching is ignored completely here, and a default action is defined for solution construction. This default action is that if no suitable solution is found in the retrieved cases, the STANDARD solution will always be adopted.

As matching figures, while comparable within an application due to the common data used, tend to be highly application specific, they will not be compared to other reasoners. However, two points can be made with regards to actual system implementation. Firstly, the higher figures for VWNN make it a more favourable technique for implementation than the three techniques preceding it. This is simply because a higher threshold for matching could be used for matching. ¿From a user perspective this might be important, as a user might tend to place little importance on low matching figures, which might lead to distrust of the reasoner. ¿From this perspective, we could view VWNN as the best of the four HACA techniques shown in the second raw of table 1. Similarly, the high matching figures found in SFVW indicates that it would also be suitable.

Number of cases retrieved can be compared to other reasoners, and here room for possible improvement is found. While the number of cases retrieved was restricted by using a threshold for retrieval of 30%, this was essentially an artificial limit, chosen for ease of use rather than through some knowledge-guided technique. In addition, these retrieval figures are still fairly high when compared to those obtained by Simoudis and Miller, where selectivity ranged between 1.5% and 3% [25]. A good concept to solve this problem might be to use domain specific knowledge to validate retrieved cases (as in Validated Retrieval - suggested in [25]), and trim the selected case set. However, this was outside the scope of the research.

Retrieval Time

Retrieval time indicates two points. Firstly, and unsurprisingly, there was negligible difference in retrieval time between the five HACA techniques. According to the first row in table 1, all of them averaged 3.2 seconds to search the case base. Since all use a nearest neighbour approach, scanning all cases in the case base individually, this similarity was not unexpected. As such, when evaluating the indexing techniques' effectiveness in relation to each other, retrieval time is not an issue.

Secondly, in relation to actual system building, with a retrieval time of 3.2 seconds for a case base of 90 cases on an older PC, simple nearest neighbour retrieval might well be a viable technique from a retrieval time point of view. Faster hardware could achieve significant speed-up. Potential speed up could be achieved through adopting a technique which shares indexes amongst cases like in GRAND [19]. However, retrieval time was not a focus of the research. While the previous discussion indicates an area for potential improvement in this research, it does not indicate a fault in the techniques implemented.

6 Conclusions and Areas for Future Research

Based on the prediction accuracies obtained, it was concluded that the case-specific indexing techniques were, as expected, the best techniques of those implemented. In addition, compared to other published results, it was concluded that the prediction accuracies obtained in the research were comparable and, hence, satisfactory. As prediction accuracy was a measure of the effectiveness of the indexing techniques, it was further concluded that the indexing techniques tested could be considered successful in the application domain chosen.

Two innovative indexing approaches were introduced and analysed in this paper. The Fixed Weight Nearest

Neighbour 80:20 (FWNN-80:20) and the Separate Field Variable Weight. Their applicability was investigated and their features were compared to the rest of the techniques.

Apart from these main conclusions, a number of areas for future research were noted. It was found, unsurprisingly, that some errors were incurred because the case base did not contain the data needed to solve the current problem. Obviously, by using only 90 of, potentially, thousands of cases, there was no hope of achieving complete coverage. Thus, an immediate area of improvement in the system would be to add more cases to the case base. Watson reports that, in order to ensure that a case base deals adequately with the majority of problems, a case base should be 80% complete before delivery for real life usage [27]

It was found during testing that while no serious errors occurred there was some inconsistency in terms of solution provision, i.e. given the same values in REPRESENTATION fields, occasionally different actions were taken. This problem could probably be attributed to different cases being handled by different consultants.

Regarding solution construction, it was found that certain expert knowledge was still not contained in the case despite the NEW fields being added since differentiation between solutions was not completely defined by the contents of the NEW fields. A further refinement of the case base is therefore necessary, either through addition of fields or through generalisation.

It was also found that even in a case base of 90 cases, there was overlap of knowledge between cases. If the system were to proceed to the use of a large database, this overlap would be undesirable, thus some sort of reorganization mechanism to remove redundant cases, as suggested in [5], might need to be considered.

A final point relates to retrieval time. To speed up performance indexes could be shared. Instead of storing indexes at a case level, indexes common to a number of cases could be stored at a higher level and matched once, thus resulting in a speed improvement.

The results of the experiments indicate that while many future avenues of research exist, the indexing techniques, designed and tested in the research as they stand, successfully perform the task of identifying the correct cases for retrieval. This result indicates the applicability of CBR to the chosen domain of pretransportation handling of hazardous waste.

7 Acknowledgements

This research was sponsored by the Foundation for Research Development and the University of Natal whose financial support is gratefully acknowledged. The authors are grateful also for the valuable suggestions by the anonymous referees and the Editor.

References

[1] A. Aamodt and E. Plaza. 'Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches'. *AI Communications*, **7**(1):39-52, (1994).

[2] D.W. Aha and R.L. Bankert. 'Feature Selection for Case-Based Classification of Cloud Types: an Empirical Comparison'. In *Proceedings of the 1994 AAAI Workshop*, AAAI Press, 106-112, (1994).

[3] A. Almonayyes. 'A multi-level Indexing Scheme for Retrieving Cases of Multiple Points of View'. In *5th German Workshop on Case-Based Reasoning (GWCBR'97) - Foundations, Systems, and Applications*, (1997).

[4] R. Barletta. 'An Introduction to Case-Based Reasoning'. *IEEE AI Expert*, **6**(8):32-49, (1991).

[5] B. Bartsch-Sporl. 'How to Introduce CBR Applications in Customer Support'. In *5th German Workshop on Case-Based Reasoning (GWCBR'97) - Foundations, Systems, and Applications*, (1997).

[6] L.K. Branting and J.D. Hastings. 'An Empirical Evaluation of Model-Based Case Matching and Adaptation'. In *Proceedings of the 1994 AAAI Workshop*, AAAI Press, 72-78, (1994).

[7] P. Buta. 'Mining for Financial Knowledge with CBR'. *IEEE AI Expert*, **9**(2):34-41, (1994).

[8] Department of Environment Affairs. 'Hazardous Waste in South Africa'. *Vol.1: Situation Analysis, Vol. 2: Technologies, Vol. 3: Policy, Vol 4: Legislative Options, Vol. 5: Impact Assessment*, Ed. R.G. Noble, CSIR, (1992).

[9] A.J. Gonzalez and R. Laureano-Ortiz. 'A Case-Based Reasoning Approach to Real Estate Property'. *Expert Systems With Applications*, **4**(2):229-246, (1992).

[10] M. Goodman. 'CBR in Battle Planning'. In *Proceedings: DARPA Workshop on Case-Based Reasoning*, Ed. K. Hammond, 264-269, (1989).

[11] U.G. Gupta. 'How Case-Based Reasoning Solves New Problems'. *Interfaces*, **24**(6):110-119, (1994).

[12] K.J. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, San Diego, (1989).

[13] J.V. Hansen, R.D. Meservy and L.E. Wood. 'Indexing and Tree Pruning Concepts to Support Case-Based Reasoning'. *Omega, International Journal of Management Science*, **22**(4):361-369, (1994).

[14] D. Hennessy and D. Hinkle. 'Applying Case-Based Reasoning to Autoclave Loading'. *IEEE AI Expert*, **7**(5):21-26, (1992).

[15] J.L. Kolodner. 'Improving Human Decision Making Through Case-Based Decision Aiding'. *AI Magazine*, **12**(2):52-68, (1991).

[16] J.L. Kolodner. *Case-Based Reasoning*, Morgan Kaufmann, San Mateo, (1993).

[17] D.B. Leake. 'CBR in Context: The Present and Future'. In *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/MIT Press, (1996).

[18] D.E. O'Leary. 'Verification and Validation of Case-Based Systems'. *Expert Systems with Applications*, **6**(1):57-66, (1993).

[19] G.D. Oosthuizen. 'A Dynamic Indexing Mechanism for Memory-Based Reasoning'. In *Proceedings of the International AMSE Conference on Intelligent Systems*, 127-136, (1994).

[20] M.J. Pazzani. 'Indexing Strategies for Goal Specific Retrieval of Cases'. In *Proceedings: DARPA Workshop on Case-Based Reasoning*, 31-35, (1989).

[21] M. Pearce, K. Goel, J.L. Kolodner, C. Zimring, L. Sentosa and R. Billington. 'Case-Based Design Support - A Case Study in Architectural Design'. *IEEE AI Expert*, **7**(5):14-20, (1992).

[22] J. Petrak, R. Trappl and J. Furnkranz. 'The Possible Contribution of AI to the Avoidance of Crises and Wars: Using CBR Methods with the KOSIMO Data Base of Conflicts'. Technical report, Austrian Research Institute for Artificial Intelligence, (1994).

[23] J.H. Pollard. *A Handbook of Numerical and Statistical Techniques*, Cambridge University Press, (1977).

[24] F.J. Radermacher, W-F Riekert, B. Page and L.M. Hilty. 'Trends in Environmental Information Processing'. In *Proceedings of the IFIP Congress 94*, Elsevier Science Publishers B.V. (North Holland), Vol. 2, 1-23, (1994).

[25] E. Seamaids and J.S. Miller. 'Validated Retrieval in Case-Based Reasoning'. In *Proceedings of AAAI-90*, Cambridge, MA:AAAI Press/MIT Press, 300-315, (1990).

[26] E. Seamaids. 'Using Case-Based Retrieval for Customer Technical Support'. *IEEE AI Expert*, **7**(5):7-12, (1992).

[27] I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufmann Publishers, San Francisco, (1997).

[28] D. Wettschereck and D. Aha. 'Weighting Features'. In *Proceedings of the First International Conference on Case-Based Reasoning (ICCBR-95)*, Springer-Verlag, Lisbon, (1995).

[29] K.L. Wortmann. 'On Case Representation and Indexing in a Case-Based Reasoning System for Waste Management'. M.Sc. thesis, University of Natal Pietermaritzburg, (1998).

Received: 21/8/98, Accepted: 26/5/99

Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications of Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript
Manuscripts for *review* should be prepared according to the following guidelines:

- Use wide margins and $1\frac{1}{2}$ or double spacing.
- The first page should include:
- the title (as brief as possible)
- the author's initials and surname
- the author's affiliation and address
- an abstract of less than 200 words
- an appropriate keyword list
- a list of relevant Computing Review Categories
- Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text according to the Harvard. References should also be according to the Harvard method.

Manuscripts accepted for publication should comply with guidelines as set out on the SACJ web page,

`http://www.cs.up.ac.za/sacj`

which gives a number of examples.

SACJ is produced using the LaTeX document preparation system, in particular LaTeX 2_ε. Previous versions were produced using a style file for a much older version of LaTeX, which is no longer supported. Please see the web site for further information on how to produce manuscripts which have been accepted for publication.

Authors of accepted publications will be required to sign a copyright transfer form.

Charges
Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30.00 per final page for contributions which require no further attention. The maximum is R120.00, prices inclusive of VAT.

These charges may be waived upon request of the author and the discretion of the editor.

Proofs
Proofs of accepted papers may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Letters and Communications
Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words. Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book Reviews
Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement
Placement of advertisements at R1000.00 per full page per issue and R500.00 per half page per issue will be considered. These charges exclude specialised production costs, which will be borne by the advertiser. Enquiries should be directed to the editor.

South African Computer Journal

Number 23, July 1999
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 23, Julie 1999
ISSN 1015-7999

Contents