Case-Based Reasoning Tools: an overview

Ian Watson
AI-CBR, University of Salford, Salford, M5 4WT, UK
ian@ai-cbr.org

Abstract

Although Case-Based Reasoning (CBR) is a relatively recent problem solving technique there are several suprisingly mature commercially available software tools . However, the numbers of people with first hand practical experience of CBR software tools is still relatively small. The main objective of this review is to provide a comprehensive review of currently available CBR software tools. The paper is the first fully independant review of currently available CBR tools on the market. It concludes with suggested improvments needed for CBR software tools.

Introduction

Despite the fact that CBR is a recent innovation there are at least ten commercially available tools with CBR functionality. The objective of this paper is to provide a comprehensive review of CBR software tools. It must be emphasised that this is an impartial review (i.e., the author has no allegence to any tool vendor). Another recent very detailed CBR tool review [Althoff et al., 95] only covers a subset of the tools reviewed here and several of its authors have close allegience to one or other of the tools reviewed. It is hoped that this review will help people identify which tool is most suitable for their needs. It is also hoped that the tool developers will both learn from each other and accept the constructive criticisms mentioned in this review.

To these ends, section two gives a brief overview of CBR as an introduction to the terminology used throughout this review (complete reviews of CBR are available in Aamodt & Plaza [94] and Watson & Marir [94]). Section three identifies a set of theoretically ideal features that a CBR tool should have. Section four describes the functionality of ten CBR software tools and present an overview in a table.. The last section summarises the review and makes two principle recommendations for CBR tool developers.

What is case-based reasoning?

> *A case-based reasoner solves new problems by adapting solutions that were used to solve old problems* [Riesbeck & Schank, 89]

At its most simple case-based reasoning is based on the observation that when we solve a problem we often base our solution on one that worked for a similar problem in the past. An example would be driving to work. When you get in the car in the morning you don’t explicitly plan your route, you take the route you usually take. If you meet a traffic jam you may remember how you avoided a similar jam in the past. If you do take an experimantal route to avoid the jam and it’s a success, you will remember it and perhaps use it again in similar circumstances in the future.

CBR is thus a deceptively simple problem solving paradigm that involves matching your current problem against problems that you have solved successfully in the past. The process can be augmented by adapting solutions so they more closely match your current problem.

The processes involved in CBR have been described as a cyclical process comprising *the four REs* Aamodt and Plaza [94]:

1. RETRIEVE the most similar case(s);
2. REUSE the case(s) to attempt to solve the problem;
3. REVISE the proposed solution if necessary, and
4. RETAIN the new solution as a part of a new case.

A new problem is matched against cases in the case base and one or more similar cases are *retrieved*. A solution suggested by the matching cases is then *reused* and tested for success. Unless the retrieved case is a close match the solution will probably have to be *revised* producing a new case that can be *retained*.

Watson, I.D (1996) Case-Based Reasoning Tools: An Overview. In, Progress in Case-Based Reasoning. Proceedings of the 2nd UK Workshop on Case Based Reasoning. Watson, I. (Ed.)University of Salford April 10th 1996. AI-CBR/SGES Publications, pp. 71-88.

A theoretically ideal feature set

A theoretically ideal CBR tool should support each of the main processes of CBR (i.e., retrieval, reuse, revision and retention). In addition, the tool should support the developers in delevering an efficient solution and the tool must integrate with other systems. Thus, we can suggest the following function set:

- **Representation -** this must support a full range of data types (e.g., numeric, string, Boolean and symbol) and should be able to structure cases in ways relevant to the application domain. Flat records of value:attribute pairs may be sufficient for some applications, but complex domains may require ordered symbol hierarchies, relationships between features and may benefit from object-oriented inheritance.
- **Retention -** the case-base should be organised into a manageable structure that supports efficient search and retrieval methods. A balance should be found between storing methods that preserve the semantic richness of cases and their indices and methods that simplify the access and retrieval of relevant cases.
- **Retrieval -** indexing of case will be necessary to make retrieval efficient so indexing must be supported. It may be automatic, but developers should be able to influence the process. If nearest neighbour is used then case features should be able to be weighted and similarity measures customised. If inductive techniques are used the index tree generated should be open to inspection and alteration by developers.
- **Revision -** this requires the provision within the tool of a programming language for case adpatation. The language may be procedural or use KBS techniques.

In addition to this essential functionality other issues are certain to be of relevance to developers, namely:

- The tool should be able to manage large case-bases with retrieval times increasing linearly (at worst) with the number of cases.
- The tool should support a variety of retrieval mechanisms and allow them to be mixed when necessary.
- The tool should provide the developer with a variety of metrics to both assist the development of an efficient system and the maintenance of the case-base.
- Since organisations may hold case data in existing databases the tool should be able to import data from the full range of corporate databases.
- A CBR tool should provide a good user interface both for the developers and for operational users.
- Since it may be necessary to embed the developed application the tool should provide a C function library (or similar e.g., a DLL) or support the use of communication protocols such as MS Windows Dynamic Data Exchange (DDE).

CBR software tools

Theoreticians might argue that the current surge in interest in CBR is due to the intuitive nature of CBR and because it may closely resemble human reasoning. Software vendors might argue that it is because CBR tools have made the theory practically feasible. There is truth in both views but certainly the tools have made a contribution. This section reviews most of the currently available major CBR tools. The tools are dealt with in alphabetical order. The section concludes with a table that summarises the functionality of the tools reviewed.

ART*Enterprise

ART*Enterprise is the latest incarnation of ART Inference Corporation’s flagship development product (currently in Version 2.0 beta). Inference Corporation[1] based in California are one of the oldest established vendors of AI tools. Inference market ART*Enterprise as an integrated, object-oriented applications development tool. designed for MIS developers offerring a variety of representational paradigms including:

- a procedural programming language
- objects supporting multiple inheritance, encapsulation and polymorphism;

---

[1] Inference Corporation has recently undergone a corporate restructuring. ART*Enterprise is now the property of a new company called Brightware. Inference controls the CBR2 family of products.

- rules; and
- cases.

This is all packaged with a GUI builder, version control facilities, and an impressive ability to link to data repositories in proprietary DBMS formats conforming to the ODBC standard for developing client-server applications. Moreover, ART*Enterprise offers cross-platform support for most operating systems, windowing systems and hardware platforms.

The CBR component in ART**Enterprise* is essentially the same as that in CBR2 (or rather vice-versa since CBR2 uses code from ART to provide its CBR functionality). This functionality is reviewed below in Section 0. However, because developers have direct access to the CBR functionality ART**Enterprise* is more controllable than in CBR2.

In conclusion, ART*Enterprise is perhaps the ideal tool for embedding CBR functionality within a corporate wide information system. Allthough the CBR functionality itself is more limited than some tools (i.e., cases are flat value:attribute pairs and there is no support for inductive indexing) the proven knowledge representational abilities of ART will make it a good tool for performing complex case adaptation. It can be assumed that since ART*Enterprise uses similar code to CBR2 that its case retrieval times will be as fast (or faster) than those recorded in Althoff et al's experiments [95]. A small word of warning is needed: although ART*Enterprise is available on the PC platform under MS Windows it is very demanding on resources - a fast 486 or Pentium with a *minimum* of 32MB of RAM is required.

(NB. ART-IM has similar CBR and knowledge representational functionality to ART*Enterprise, but lacks the data integration, GUI builder and version control features).

Case-1

Case-1 is a new CBR tool from Astea International (a beta release of Version 1.0 was reviewed). The company has a background in providing integrated sales, support and service systems. Case-1 was obviously developed with CBR Express in mind and it shares many of its features. Cases are represented as free form text describing a problem, a set of weighted questions that can confirm or reject a case and a set of solutions. As with CBR2 cases can be authored by people who have no programming experience. Cases are stored in a relational database (Watcom) and the interface is developed using Visual Basic (Case-1 runs under MS Windows). Case-1 does score over CBR Express in letting case authors have easy access to the lexicon of words ignored during text matching, however the product is not as mature as CBR2 and does not seem to offer any significant functional improvements. However, the tool is well integrated with Astea's other customer support tools and therefore if you are already a client of Astea you may be advised to use Case-1.

CasePower

Formerly called Induce-it from Inductive Solutions Inc. CasePower builds its cases within the spreadsheet environment of Microsoft Excel. CasePower is a specialised tool for constructing Excel spreadsheets that can be analysed usng CBR. Within the limited confines of Excel it provides basic CBR functionality mainly suitable for numeric applications. Symbolic data can be represented as ordered hierarchies that are mapped to numerical ranges. However, for more complex non-numerical applications another CBR tool may be preferable.

CasePower uses nearest neighbour retrieval and it reduces search time by calculating an index for each case in advance. This can be a lengthy process for a large case-base but it does reduce retrieval times. The system simply calculates the index for the new case and compares it against the pre-calculated indices of the case-base. If a new case is to be retained, the entire set of case indices must be recalculated. Adaption can be performed using Excel formulae and macros. Similarly all the other features of Excel are available such as graphing, reporting and DDE.

CBR2 (CBR Express, CasePoint, Generator & Tester)

Produced by Inference Corporation, the CBR2 family of products are certainly the most successful CBR products to date with over 500,000 licences sold world-wide. CBR2 is specifically tailored to the vertical market of customer support help desks.

The CBR2 family of tools having the following roles:

- **CBR Express** is a development or authoring environment for cases, it also features a customer call tracking module.
- **CasePoint** is a search engine for case-bases developed using CBR Express.

- **Generator** is a tool that automates the creation of cases-bases from sets of MS Word or ASCII files.
- **Tester** is a tool that provides a variety of metrics for case-base developers using CBR2

CBR2 uses a simple case structure of flat records. Cases comprise a title, a description, a set of weighted questions (effectively value:attribute pairs), and a set of actions. Cases can be stored in almost any proprietary database format, although the Raima database format is supplied as a default. CBR2 is network ready and case-bases can be shared across an organisation's network

CBR2 uses nearest neighbour matching to initially retrieve cases by matching a users free text query against the title and descriptions of cases in the case-base. A key feature of CBR2 is its ability to handle free-form text. This was felt to be vital to the help desk market since it lets customers describe their problems in their own words rather than being taken through a decision tree style question and answer session. CBR2 ignores words such as: and, or, I, there, etc., it can use synonyms, and represents words as a set of trigrams. The trigram for cartridge is: CAR, ART, RTR, TRI, RID, IDG, DGE. The use of trigrams means that CBR2 is very tolerant of spelling mistakes and typing errors such as letter transpositions. The trigrams for cartridge and cartrideg will still match closely. Although there are obvious problems with this lexical approach it is nonetheless surprisingly powerful and very useful for CBR2's market.

After an initial set of relevant case are retrieved using the textual matching retrieval then becomes knowledge guided as questions are be asked to focus case retrieval. Developers using CBR2 use an interface (CBR-Express) that deals with all programming elements of case creation and editing resulting in a syntax free environment that lets people without programming experience quickly develop case-bases [Watson & Abdullah, 94].

The interface of CBR Express is constructed using Asymetrix ToolBook version 1.5, and developers with the authoring version of ToolBook can obtain access to the source code of the interface to customise it. However, this is a non-trivial task and should only be attempted by experienced ToolBook developers. Otherwise there is a real risk of compromising the functionality of the system (this advice also applys to KATE, see Section 0)

During retrieval CBR2 examines a user's free form text entry and matches this against cases' titles and descriptions. This results in the retrieval of a set of cases. A list of ranked solutions with likelihood values is generated from the cases and the user is offered these along with a set of questions. Answers to these questions help narrow the number of cases that match leading to a more accurate solution that is presented to the user. In the event of a solution not being reached (CBR2 has a customisable threshold value) or if a solution is not satisfactory the CBR cycle is closed by using the concept of an *unresolved* case (CASE-1 has borrowed this concept from CBR2). An unresolved case saves the entire transcript of the consultation so the case-base administrator can subsequently find out what that case's solution was and modify the unresolved case to create a new case.

If you want to integrate or embed CBR2 it is probably more efficient to use CasePoint as a DDE server application (it is also now available as a DLL). As a delivery vehicle CasePoint has many advantages over CBR Express. A criticism of nearest neighbour matching is that if a case base were large and if cases had many features it is not an efficient process. However, the matching algorithm that CasePoint uses is extremely fast. CasePoint also supports the use of a rule-file that identifies keywords in the query text and automatically answers certain questions. It can also order questions so that they best discriminate between cases under consideration.

Eclipse

From Haley Enterprises Eclipse is a close relative of ART. The forward chaining functionality of ART, written in LISP, was re-implement in C by NASA, entering the public domain as the language CLIPS. In the late eighties Paul Haley, the former Chief Scientist of Inference, developed a new ART-like language compatible with CLIPS. This became Eclipse. Like ART, Eclipse offers objects, only this time fully compatible C++ objects, and optimised forward chaining using the Rete algorithm. Eclipse is available for the DOS operating system, MS Windows, UNIX platforms and certain mainframe environments.

The Easy Reasoner is a module within Eclipse that offers CBR functionality similar to that of the Inference products. Eclipse is only available as a C library (i.e., there is no development interface) and is therefore only suitable for experience C programmers. Eclipse supports the usual range of variable types and offers similar text handling facilities to ART (i.e., ignoring noise words and using trigrams to cope with spelling mistakes). Interestingly Eclipse also uses stems to identify, for example, that *magnetically* and *magnetic* all stem from *magnet*. Once cases have been retrieved they can be asserted as Eclipse objects for adaptation by its rule-base.

ESTEEM

ESTEEM, from Esteem Software Inc., was originally in written in Intellicorp's Kappa-PC. Version 1.4 is now written in C++ and has its own inference engine enabling developers to create adaptation rules. It supports case hierarchies that help narrow the search. It also supports applications that access multiple case-bases and nested cases. This means that one can reference another case-base through an attribute slot in a case. ESTEEM also provides control of the induction process (ID3) through feature counting, weighted feature computation, inferred computation. Nearest neighbour matching is also supported.

ESTEEM runs on PC Windows and represents exceptionally good value for money ($495).. The developers nterface comprises five simple editors that define cases, customise similarity assessment and retrieval, define adaptation rules, import data from ASCII files of databases, and create simple form-based user interfaces. Version 1.4 now supports multimedia as a feature type and can be used as a MS DDE server for application embedding.

The source Kappa-PC (KAL) code is available from ESTEEM and thus lets developers embed CBR functionality within the Kappa environment. Moreover, Kappa-PC code can be exported as C code which can then be compiled (using a C compiler) into a stand alone .EXE file.

KATE

KATE, produced by Acknosoft in Paris, is made up of a set of tools sometimes referred to as CASECRAFT (i.e., KATE-INDUCTION, KATE-CBR, KATE-EDITOR and KATE-RUNTIME. Development should be on PC Windows (as the interface components are made with ToolBook) but deployment can be on PC Windows, Mac or SUN.

KATE-INDUCTION is an ID_3 based induction system that supports an object representation for cases. Cases can be imported from many database and spreadsheet formats. The induction algorithm is very tolerant of missing data and can make use of background knowledge. Retrieval using trees generated by induction algorithm is extremely fast [Althoff et al. 95].

KATE-CBR is the nearest neighbour component of the suite. Users can customise similarity assessments and snce it supports the same object hierachies as KATE-INDUCTION the two techniques can be combined.

KATE-EDITOR is a set of C DLL's that are integrated with ToolBook to form a customisable developers interface. In particular easy forms can be developed to assist case entry.

KATE-RUNTIME is another set of interface utilities that can be customied with ToolBook to deliver an application. KATE can also be delivered as embedded C code.

The KATE tools are a powerful set of well integrated CBR tools. Retrieval is extremly fast (even with large case bases) and can be customised by experienced developers. KATE is one of the few tools to include automatic testing routines.

ReCall

ReCall is a CBR trademark of the Paris based AI company ISoft. This tool offers a combination of nearest neighbour and inductive case retrieval. ReCall is coded in C++ and is available on the PC under Windows 3.1, on UNIX Workstations under Motif, for: SUN, IBM RS6000, BULL DPX20, HP series 700, and DEC Alpha. It is designed on an open architecture allowing users to add CBR functionality to their own applications.

Recall presents an object-oriented language with taxonomies, inheritance and multiple-inheritance mechanisms, typed descriptors, facets, deamons, and relationships between objects (individual cases are represented as instances). This allows users to specify complex domain knowledge in a structured but modular way, and to describe cases having noisy, incomplete and uncertain descriptions since feature values can be inherited. Recall provides multiple hierarchical indices that are used for organisation purposes and for efficient case retrieval. ReCall provides different methods for automatically analysing the case base providing for selection of indices as well as their organisation. However, experienced developers can impose their own organisation. Automatic procedures are based on inductive techniques. The automatic procedures takes into account the domain knowledge defined in the cases, helping users to develop applications interactively. Similarity functions take into account both the properties and values of descriptors, as well as structural differences between cases. ReCall uses a variant of a nearest-neighbour algorithm that improves similarity computations.

ReCall supports two different adaptation mechanisms: a default adaptation mechanism based on a voting principal, and user defined adaptation rules. As ReCall is based on C++, external function calls can provide more complex adaptation. ReCall can be interfaced to external applications in particular with data bases and since ReCall is available as a C++ library, CBR functionality can be integrated with other applications.

Through the use of specialised graphic editors, the developer can define objects, relationships between objects, taxonomies, deamons and adaptation rules. A tree editor allows the user to interact directly on the case organisations in order to control and modify indices. A user mode allows developers to write adaptation rules or daemons, whilst a developer mode gives access to an interpreted language.

ISoft have also recently released a product called AC^2 that uses case representation and induction facilities of ReCall for data minning.

ReMind

Produced by Cognitive Systems Inc. ReMind was developed with support from the US DARPA programme. It was originally developed for the Macintosh and has since been ported to MS Windows and some UNIX platforms.

ReMind offers template, nearest neighbour, inductive, and knowledge-guided inductive retrieval. The template retrieval supports simple SQL-like queries returning all cases that fall within set parameters. The nearest neighbour retrieval is informed by user defined importance weightings that can be placed on case features. Inductive retrieval can be done automatically by ReMind with no user involvement or the user can create a qualitative model to guide the induction algorithm (based on CART).

Qualitative models (Q-models) are created graphically to indicate which concepts (case features) are dependent on other concepts. Qualitative weightings can be placed on these dependencies and ReMind then uses the Q-model to guide the induction algorithm (hence knowledge-guided induction) resulting in decision trees that more closely reflect the causal relationship of concepts in the cases. Interestingly, different qualitative models can be created to explore different theories about the domain or to allow what-if questions to be asked.

In ReMind case adaptation is provided by creating adaptation formulae that adjust values based on the difference between the retrieved and the new case. These are also created graphically using a visual programming technique. Although this takes a little getting used to the extremely close typing of case features combined with the close typing of the operators does reduce syntax errors.

ReMind is available as a C library, for embedding in other applications, and as an interactive development environment. ReMind is a flexible CBR tool offering a wide range of case-retrieval methods along with interesting concepts such as Q-models and visual adaptation formulae. It does not have the powerful text handling features of Inference’s products and Eclipse, though it does provide an elementary natural language capability via a lexicon of terms that can be mapped to an ordered symbol hierarchy. However, in general users are forced to select rather than describe a situation. ReMind is perhaps the most flexible of CBR tools currently on the market. In particular the use of background knowledge in the form of Q-models and prototypes to inform the induction algorithm can greatly improve the efficiency of cluster trees. ReMind, however has two major limitations:

- cases are stored in an entirely hidden way; they cannot be exported or viewed by any other application;
- the nearest neighbour alorithm in ReMind is slow, and is only suitable for small case-bases. Inductive retrieval is very much faster, but building a large cluster tree is extremely slow taking several hours for a several thousand cases.

S_3-Case

S_3-Case is part of the German company techInno’s S_3 environment for systems maintenance running on PC Windows, Mac, OS/2 and various UNIX platforms. Written in SMALLTALK it supports an object oriented model with inductive (ID_3) and nearest neighbour retrieval and adpatation via forward chaining inference engine. Rule can also be used to prune the search space before retrieval. A simple user interface can be customised to suit user needs. Experienced SMALLTALK developers can embed or extend the functionality of S_3-CASE.

Conclusion

All the tools described above differ significantly. Inference’s CBR2 family of tools stand out for their maturity and robustness and are highly recommended for customer support desks (their intended market). However, beyond that application area they are limited. The same limitations apply to the less mature support desk tool CASE-1. Of the more powerful tools ESTEEM has a good functional specification and is excellent value for money. ReMind is a powerful and flexible tool and is notable for features such as Q-Models and visual adaptation formulae. Many of the tools reviewed (e.g., Eclipse, KATE, ReCall, and ReMind) are avialble as C libraries and can be embedded. However, this is only possible for experienced programmers. The use of communications protocols such as DDE is welcomed since it can more easily be used by less experienced programmers (e.g., CasePower, CBR2, and ESTEEM).

Most of the tools reviewed meet the necessary set of features identified in section 0. However, none fully meets all of the desirable features. CBR2 is the only product to take case-base metrics seriously through its Tester module (this is discribed in more detail in Watson 95) although ReMind does offer some case-base statistics whilst KATE can perform some automatic tests. Many of the tools now offer some data integration but ART*Enterprise offers the best data integration along with excellent adaptation facilities through its powerful knowledge representation and programming environment.

To conclude, no single tool is perfect, though there is probably one that will suit most of your requirements and your budget. Developers of new CBR tools must realise that CBR applications will not always be stand alone; they must to be able to import records from existing databases and communicate with other applications. Most importantly, developers need better support in building efficient case-bases. Hence, they need to be provided with metrics on case structure, and on the efficiency and performance of retrieval mechanisms. It is perhaps no suprise that Inference, with its huge user base, is the first company to fully realise this and has consequently developed the Tester module for CBR2.

This work was partially funded by EPSRC project number GR/J42496. The author would like to thank all the software companies that gave evaluation copies of their software and information about their products.

Product ***Vendor***	**Platforms**	**Representation**	**Retrieval**	**Adaptation**	**Interface**	**Other Comments**	**Price (approx.)**
ART*Enterprise *Inference Corp*.	a wide variety of PC, workstation, DEC, IBM HP and mainframe environments	flat value attribute pairs supporting a full range of variable types	nearest neighbour but can be augmented using ART's programming environment	functions, rules and other knowledge-based techniques	fully featured GUI builder	excellent data integration with most DBMS formats and version control	PC version $12,000
CASE-1 *Astea International*	PC Windows	flat records supporting text and weighted questions	nearest neighbour and knowledge-guided retrieval	no adaptation features	interface cannot be customised	designed for help desks Version 1.0 is now shipping	???
CasePower *Inductive Solutions Inc*.	PC Windows, Macs OS/2	MS Excel Spreadsheet ordered symbol hierachies and nested cases	nearest neighbour	via Excel functions and macros	Excel interface	Excel must be bought to use this product	less than $1000
CBR2 *Inference Corp*.	PC Windows and MVS version	flat records supporting text and weighted questions	nearest neighbour and knowledge-guided retrieval	no adaptation features	ToolBook interface of CBR-Express can be customised	runtime, testor and generator modules available	PC version CBR Express $10,000
Eclipse *The Haley Enterprise*	any ANSII C environment	flat value attribute pairs full range of variable types	nearest neighbour	functions, rules and other knowledge-based techniques	no interface it is only supplied as a C library	this ART-like product is very fast	???
ESTEEM *Esteem Software Inc*.	PC Windows UNIX/X Motif	cases can be ordered hierarchically and can be nested	nearest neighbour and inductive (ID$_3$)l retrieval	functions and rules	GUI Builder	now supports DB access and multimedai	$495
KATE *AcknoSoft*	PC Windows & UNIX	hierachical cases	nearest neighbour and induction (ID$_3$)		ToolBook interface can be customised	available as a C library for embedding	???
ReCall *ISoft*	PC Windows & UNIX	hierachical cases with relationships	nearest neighbour and induction	deamons	graphical development environment	available as a C++ library for embedding	$9,000
ReMind *Cognitive Systems Inc*.	PC Windows Mac & UNIX	nested flat cases & ordered symbol hierarchies	nearest neighbour, induction (CART) & template retrieval	formulae	development interface can be customised	available as a C library for embedding	$6,000
S$_3$-CASE tecInno GmbH	PC Windows Mac & UNIX	object oriented hierarchical cases	nearest neighbour and induction (ID$_3$)	rules	customisable interface	written in SMALLTALK	???

References

Aamodt, A. & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, 7(i): pp 39-59.

Althoff, K-D, Auriol, E., Barletta, R. & Manago, M. (1995*). A Review of Industrial Case-Based Reasoning Tools*. AI Intelligence, Oxford, UK. ISBN 1 898804 01 X

Kolodner, J. L. (1993). *Case-Based Reasoning*. Morgan Kaufmann.

Reisbeck, C.K, & Schank, R.C. (1989). *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Hillsdale, NJ, US.

Watson, I.D., & Abdullah, S. (1994). *Developing Case-Based Reasoning Systems: A Case Study in Diagnosing Building Defects*. In, Proc. IEE Colloquium on Case-Based Reasoning: Prospects for Applications, Digest No: 1994/057, pp.1/1-1/3.

Watson, I. & Marir, F. (1994). *Case-Based Reasoning: A Review*. The Knowledge Engineering Review, Vol.9 No.4 pp.327-54.

Suppliers

*AknoSoft, **KATE***
58a, Rue du Dessous des Berger
75013 Paris, France
Tel: (33-1) 44 24 88 00
Fax:(33-1) 44 24 88 66
*Esteem Software Inc,. **ESTEEM***
302E. Main street Cambridge City,
IN 47327, USA
Tel: (317) 478-3955
Fax:(317) 478-35550
*Inference Corporation, **ART*Enterprise**,**CBR 2**.*
550 North Continental Blvd, El Segundo
California 90245, USA

*The Haley Enterprise Inc, **Eclipse***
413 Orchard Street
Sewickley, PA USA 15143
Tel: (412) 741-6420
Fax: (412) 741-6457

Astea International, CASE-1
55 Middlesex Turnpike
Dedford, MA 01730, USA
Tel: (617) 275-5440
Fax: (617) 275-1910

*Cognitive System Inc., **ReMind***
220-230 Commercial Street, Boston, MA 02 109, USA.
Tel: (617) 742-7227
Fax: (617) 742-1139.
*Inductive Solution Inc., **CasePower***
380 Rector Place, Suit 4A,
New York, NY 10280, USA
Tel: (212) 945-0630
Fax:(212) 945-0367
*Isoft, **ReCall***
Chemin de Moulon
F-91190 Gif sur Yvette
France
Tel: (33-1) 69 41 27 77
Fax: (33-1) 69 41 25 32
*tecInno GmbH, **S_3-CASE***
Sauerwiesen 2
67661 KAISERSLAUTERN
Germany
Tel: +49 6301-60-60
Fax: +49 6301-60-666

Information on all aspects of case-based reasoning can be found at www.ai-cbr.org