Watson, I. & Gardingen, D. (1999). *A Distributed Case-Based Reasoning Application for Engineering Sales Support*. In, Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99), Vol. 1: pp. 600-605. Morgan Kaufmann Publishers Inc. ISBN 1-55860-613-0
(this paper was awarded an IJCAI-99 Distinguished Paper Award)

A Distributed Case-Based Reasoning Application for Engineering Sales Support

Ian Watson
AI-CBR
University of Salford
Salford, M7 9NU
UK
ian@ai-cbr.org
www.ai-cbr.org

Dan Gardingen
Western Air Ltd.
Fremantle
Western Australia

Abstract

This paper describes the implementation of a distributed case-based reasoning application that supports engineering sales staff. The application operates on the world wide web and uses the XML standard as a communications protocol between client and server side Java applets. The paper describes the distributed architecture of the application, the two case retrieval techniques used, its implementation, trial, roll-out and subsequent improvements to its architecture and retrieval techniques using introspective reasoning to improve retrieval efficiency. The benefits it has provided to the company are detailed.

1 Introduction

Western Air is a distributor of HVAC (heating, ventilation and air conditioning systems in Australia with a turnover in 1997 of $25 million (US dollars). Based in Fremantle the company operates mainly in Western Australia, a geographic area of nearly two million square miles. The systems supported range from simple residential HVAC systems to complex installations in new build and existing factories and office buildings.

Western Air has a distributed sales force numbering about 100. The majority of staff do not operate from head office but are independent, working from home or a mobile base (typically their car). Until recently, sales staff in the field would gather the prospective customer's requirements using standard forms and proprietary software, take measurements of the property and fax the information to Western Air in Fremantle. A qualified engineer would then specify the HVAC system. Typically the engineer would have to phone the sales staff and ask for additional information and the sales staff would have to make several visits to the customer's building and pass additional information back to the head office engineer.

Western Air felt that basing a quote on the price of a previous similar installation gave a more accurate estimation than using prices based on proprietary software, catalogue equipment prices and standard labour To try to help engineers make use of all the past installations a database was created to let engineers search for past installations. The database contained approximately 10,000 records, each with 60 fields describing the key features of each installation and then a list of file names for the full specification. Initially the engineers liked the database and it increased the number of past installations they used as references. However, after the honeymoon ended, they started to complain that it was too hard to query across more than two or three fields at once. And that querying across ten or more fields was virtually impossible. In fact most of them admitted to using the database to laboriously browse through past installations until they found one that looked similar to their requirements.

2 Implementation

Western Air decided that merely improving the efficiency of the engineers in Fremantle would not solve the whole problem. Ideally they would like the sales staff to be able to give fast accurate estimates to prospective customers on the spot. However, they were aware that there was a danger that the less knowledgeable sales staff might give technically incorrect quotes.

2.1 The Team

The development team comprised:

- a senior engineer from Western Air (one of the firms owners) as project champion,
- an engineer from Western Air to act as project manager and domain expert,
- a consultant Java/HTML programmer,
- a consultant from AI-CBR to advise on CBR issues (resident in the UK), and
- a part-time data entry clerk.

2.2 Implementation Plan

The project had the direct involvement of one of the firms owners so management commitment was not a problem. It was also decided that creating a partially functional prototype was not sensible since the system would either work or not. However. a carefully controlled and monitored trial was considered essential for two reasons:

1. It was still not certain that sales staff could create technically sound first estimates and therefore a small carefully monitored trial was essential to avoid losing the firm money.
2. There were resource implications since although all sales staff had portable PCs, some were old 486 Windows 3.1 machines and few had modems or Internet accounts.

A fixed (non-negotiable) budget was given to the project of $32,000 and it was decided that six months would be given for development and trial of the system. The project started in October of 1997 and the trial was planned for March of 1998.

It was decided initially to deal with moderately complex residential HVAC systems because it was felt that this would provide a reasonable test of the system without undue risk. Western Air felt that it was commercially unwise to risk experimentation on high value commercial contracts. Western Air realised they wanted a system that could find similar installations without making the query too complex for the engineers. Web based CBR applications have been demonstrated for a few years now such as the FAQFinder and FindME systems [Hammond et al., 1996] and those at Broderbund and Lucas Arts [Watson, 1997].

The solution they envisaged was to set up a web site that sales staff could access from anywhere in the country. Through a forms interface the prospect's requirements could be input and would be passed to a CBR system that would search the library of past installations and retrieve similar installations. Details of the similar installations along with the FTP addresses of associated files would then be available to the sales staff by FTP. The sales staff could then download the files and use these to prepare an initial quote. All this information would then be automatically passed back to an engineer to authorise or change if necessary. Once an installation was completed its details would be added to the library and its associated files placed on the ftp server.

Since a simple nearest neighbour retrieval algorithm would suffice implementing our own system was a viable option. Java (Visual Café) was chosen as the implementation language for both the client and server side elements of the CBR system. XML (eXtensible Markup Language) [WWW Consortium, 1997] was used as the communication language between client and server-side applets. The World-Wide Web Consortium (W3C) finalised XML 1.0 in December 1997 as a potential successor to HTML [WWW Consortium].

2.3 System Architecture

On the sales staff (client) side a Java applet is used to gather the customer's requirements and send them as XML to the server. On the server side another Java applet (a *servlet*) uses this information to query the database to retrieve a set of relevant records. The Java servlet then converts these into XML and sends them to the client side applet that uses a nearest neighbour algorithm to rank the set of cases. A schematic of the architecture is shown in Figure 1.

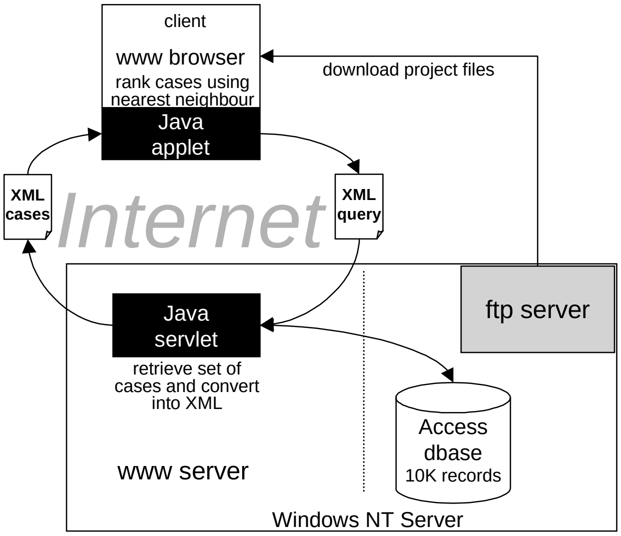


Figure 1. System Architecture

2.4 Case Representation

Cases are stored within a database. Each record (case) comprises 60 fields used for retrieval and many more used to describe the HVAC installations. In addition, links to other files on the FTP server are included to provide more detailed descriptions. Once retrieved from the database the records are ranked by a nearest neighbour algorithm and dynamically converted into XML for presentation to the client browser. An XML case representation is used by our system [Shimazu, 1998]. XML pages can contain any number of user defined tags defined in a document type definition (DTD) file. Tags are nested hierarchically from a single root tag that can contain any number of child tags. Any child tag in turn can contain any number of child tags. Each tag contains a begin statement (e.g. <Case>) and an end statement (e.g. </Case>). This is illustrated in Figure 4.

2.5 Case Retrieval

Case retrieval is a two stage process. In stage one the customer's requirements are relaxed through a process of *query relaxation*. This process takes the original query and relaxes terms in it to ensure that a useful number of records are retrieved from the database. This is similar to the technique used by Kitano & Shimazu [1996] in the SQUAD system at NEC, although as is discussed in section 3, we have improved it efficiency using an introspective learning heuristic.

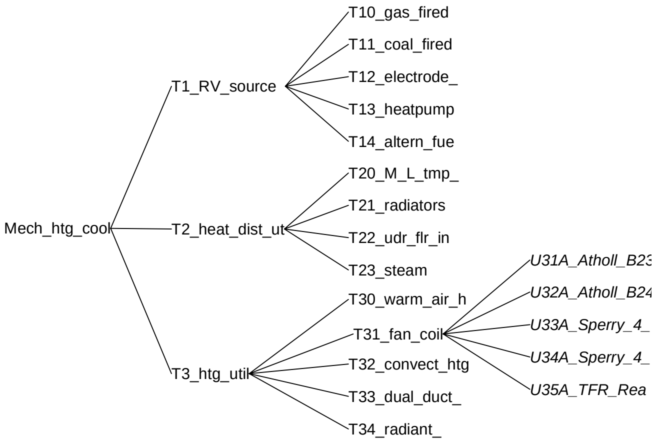


Figure 2. A Portion of a Symbol Hierarchy for Mechanical Heating & Cooling Systems

For example, assume we are trying to retrieve details of installations using Athol B23 equipment. An SQL query that just used “Athol_B23” as a search term might be too restrictive. Using an ordered symbol hierarchy (as in Figure 2) our system knows that “Athol B23” is a type of “Fan Coil” system so the query is relaxed to “Where (((EquipmentReference) = “T31_fan_coil””)..)). This query will include equipment from Athol, Sperry and TFR. An ordering of each set of symbols in the hierarchy is obtained through the reference number suffixes to each symbol (e.g. T10, T11, T12, T13, T14 as shown in Figure 2). The symbol hierarchies are stored in a table in the database.

Other specific criteria, elevations or temperatures that are numbers (integers or reals) can be relaxed by using simple ranges (e.g. a temperature of 65° F. could be relaxed to “Between 60 And 70”). Knowledge engineering was required to determine by what amounts numeric features should be relaxed. The relaxation is expressed as a term ± a percentage (e.g., “Relax_Temp = ± 10%”). These relaxation terms are stored in a table in the database. An example of a relaxed SQL query is shown in Figure 5.

In the second stage of retrieval the small set of retrieved records are compared by the client-side applet with the original query and similarity is calculated using the simple nearest neighbour algorithm shown in Figure 3. The resulting similarity measure is normalised to give a percentage range of 0% (i.e. completely dissimilar) to 100% (i.e. completely similar). The weighting on the features by default is set to 1 (i.e., all features are by default considered of equal importance) However, the sales engineers can change the feature weightings to reflect client priorities or their own preferences.

$$Similarity(T,S) = \sum_{i=1}^{n} f(T_i, S_i) \times w_i$$

where:
T is the target case
S is the source case
n is the number of features in each case
i is an individual feature from 1 to *n*
f is a similarity function for feature *i* in cases *T* and *S* and
w is the importance weighting of feature *I*

Figure 3 The Nearest Neighbour Algorithm

Once an HVAC installation is completed its details are added to the database and its associated files placed on the FTP server. Having a database management system for the case repository has proved essential since it makes it easier to generate management reports and ensure data integrity. It would be almost impossible to maintain a collection of 10,000 cases without a DBMS.

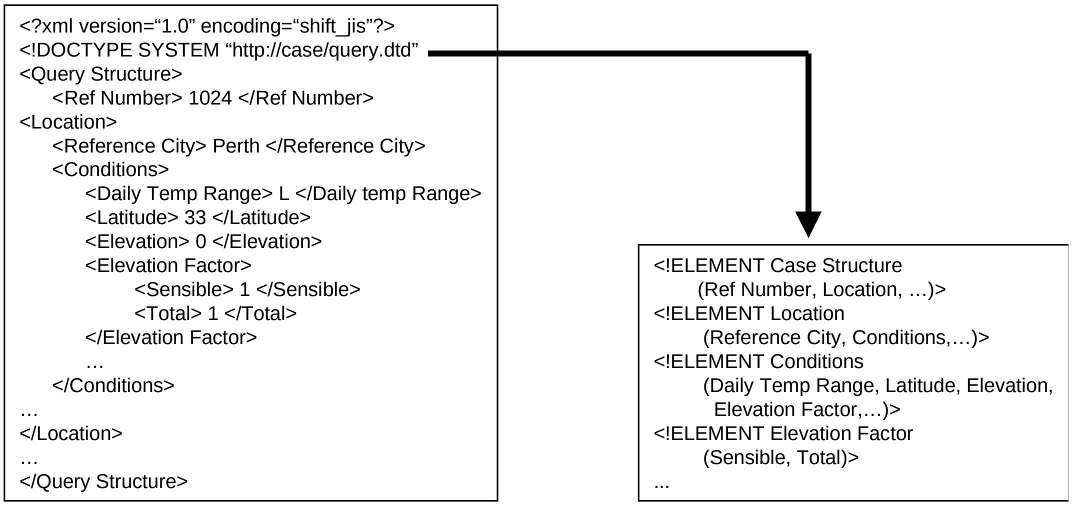
2.4 Interface Design

The interface to the system is a standard Java enabled web browser (Netscape or Internet Explorer). The forms within the Java applet were designed to look as similar to the original forms, HVAC specification tools and reports that the sales staff were already familiar with. Microsoft FrontPage 98 and Macromedia’s DreamWeaver were the primary tools used to create the web site. A sample screen from the client side Java applet can be seen in Figure 6.

2.4 Testing, Roll Out & Benefits

Two weeks before trial five test scenarios were created that were representative of the range of more complex residential installations the system would be expected to handle in use. These were given to the five sales staff who would initially use the system and they were asked to test the system. Out of the 25 tests (5x5) 22 were correct. Although the remaining three were not specified as expected they were felt to be technically acceptable solutions.

The system was rolled out for trial to the five sales staff in March of 1998. Acceptance of the system from the five sales staff was very good once they understood what it was doing. During the month’s trial the system dealt with 63 installations all of which were felt to be technically sound. The sales staff had not had to use the expertise of the HVAC engineers at all for this work although the engineers checked the final specifications.



```
<?xml version=“1.0” encoding=“shift_jis”?>
<!DOCTYPE SYSTEM “http://case/query.dtd”
<Query Structure>
    <Ref Number> 1024 </Ref Number>
<Location>
    <Reference City> Perth </Reference City>
    <Conditions>
        <Daily Temp Range> L </Daily temp Range>
        <Latitude> 33 </Latitude>
        <Elevation> 0 </Elevation>
        <Elevation Factor>
              <Sensible> 1 </Sensible>
              <Total> 1 </Total>
        </Elevation Factor>
        …
    </Conditions>
…
</Location>
…
</Query Structure>
```

query.xml

```
<!ELEMENT Case Structure
      (Ref Number, Location, …)>
<!ELEMENT Location
       (Reference City, Conditions,…)>
<!ELEMENT Conditions
       (Daily Temp Range, Latitude, Elevation,
         Elevation Factor,…)>
<!ELEMENT Elevation Factor
       (Sensible, Total)>
...
```

query.dtd

Figure 4. A Sample of the XML Case Description

```
SELECT Location.ReferenceRegion, Location.DailyTempRange, Location.Lattitude, Location.Elevation, Lo-
cation.ElevationFactorS, Location.ElevationFactorT, Location.DryBulbTempWin, Location.DryBulbTempSum,
Location.WetBulbTemp,
…
FROM Location
WHERE (((Location.ReferenceRegion)="SW") AND ((Location.Elevation) Between 0 And 100) AND ((Loca-
tion.DryBulbTempWin) Between 50 And 60) AND ((Location.DryBulbTempSum) Between 60 And 70))
…
```

Figure 5. Example of an SQL Query That Has Been Relaxed

During the trial month the five sales staff were able to handle 63 installation projects without having an HVAC engineer create the specification. This resulted in a considerable saving in engineers time allowing them more time to deal with complex high value commercial HVAC contracts.

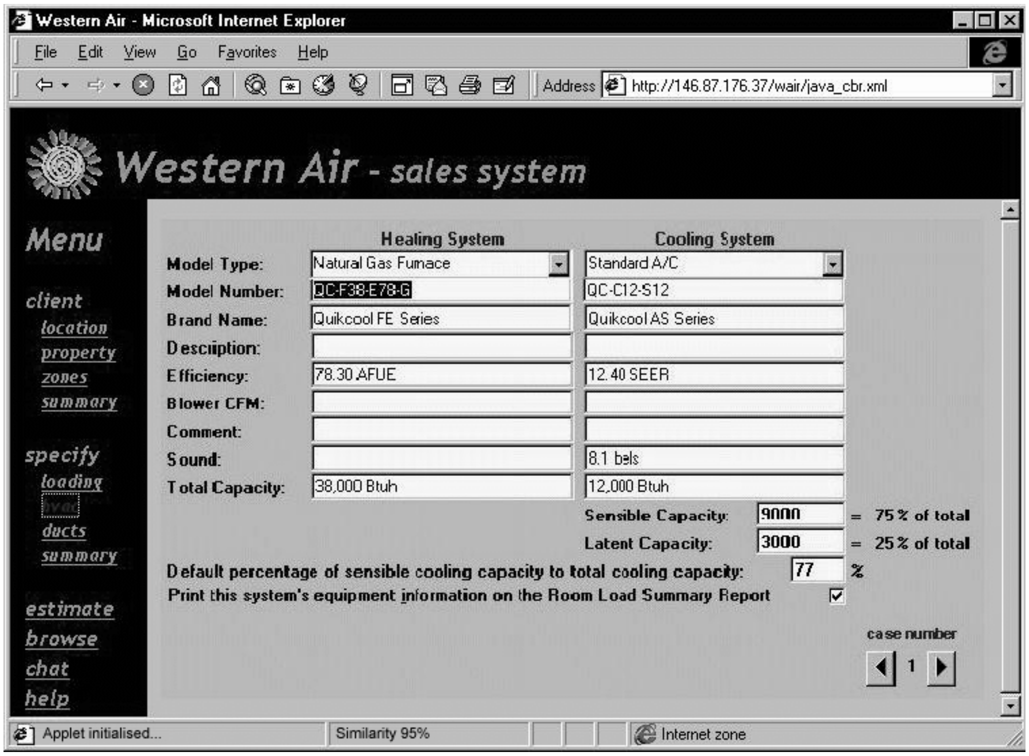
It was estimated that margins had been increased by nearly 2% while still remaining competitive. Based on this Western Air has invested $127,000 in purchasing Pentium notebook PCs for its sales staff. The system was rolled out to the entire sales staff in May of 1998. In the first ten months since the system has been fully operational sales compared with the same period last year are up 10% and margins have increased by 1.75%. Western Air expect the system to make $476,000 in its first year. A return of $253,855 on an investment of $222,145 (all amounts are in US dollars).

One of the firm’s senior engineers commented that: “*Since this system went live I’ve had much more time to spend on my own contracts. I used to hate going into the office because I always had a string of problems to handle from the mob out in the field. Now I feel I have the time to really help when I do get a problem to deal with.*”

A member of the sales engineering staff said that: “*This is just great. It used to be really frustrating waiting for them back in Fremantle to deal with our problems. I always had to give ‘em aggro and when we did finally get an answer the bloody customer changed his mind. Then they whinge because we can’t give them an answer on the spot. Now I can even use their phone and get good answers real quick. It really impresses them!*”

3 System Enhancements

Since its roll-out the original implementation of the system has experienced increasing load performance problems. The Java servlet approach suffered from poor performance because the web server loads, executes and terminates a new servlet program for each user access. Large data sets and complex queries especially burden the system because data querying takes place via the Java servlet program rather than directly via the database. This coupled with the fact that MS Access is not a particularly fast database caused time out problems as the server load increased. To rectify this problem the database was ported to mySQL (http://www.tcx.se) a freeware database with much better performance.


Western Air - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address http://146.87.176.37/wair/java_cbr.xml
Western Air - sales system
Menu
client
location
property
zones
summary
specify
loading
hvac
ducts
summary
estimate
browse
chat
help
case number
Applet initialised...
Internet zone

	Heating System	Cooling System
Model Type:	Natural Gas Furnace	Standard A/C
Model Number:	QC-F38-E78-G	QC-C12-S12
Brand Name:	Quikcool FE Series	Quikcool AS Series
Desciption:		
Efficiency:	78.30 AFUE	12.40 SEER
Blower CFM:		
Comment:		
Sound:		8.1 bels
Total Capacity:	38,000 Btuh	12,000 Btuh

Sensible Capacity: 9000 = 75% of total
Latent Capacity: 3000 = 25% of total

Default percentage of sensible cooling capacity to total cooling capacity: 77 %

Print this system's equipment information on the Room Load Summary Report ☑

case number 1

Similarity 95%

Figure 6. Client Side Java Applet Showing Retrieved HVAC Case

In addition Netscape's LiveWire database integration tool was used. This product has excellent database query functions, and importantly, because the LiveWire engine runs within the Netscape Web Server process it can share database connections across all Web accesses.

3.1 Introspective Learning

The initial query relaxation method of first performing a precise query and then relaxing the query through successive iterations until a sufficiently large set of cases was retrieved also compounded the performance problems. A suggestion was made to turn this process around – namely, why not relax the initial query far enough to ensure that a large set of cases would be retrieved (e.g., several hundred cases) and then refine the query to reduce the sub-set to around twenty cases). The obvious speed advantage in this approach is that an increasingly small sub-set of cases is searched in any subsequent iterations as opposed to the entire database in the original query relaxation approach.

However, deciding how much to relax the query was not straightforward so an introspective learning approach was taken. This is an approach in CBR where the reasoning system itself learns over time to modify its internal representation to improve its performance [Markovitch & Scott, 1993]. For example, CBR systems may learn to modify feature weights, adaptation rules [Leake, et al., 1995; Hanney & Keane, 1996], or even learn to forget redundant cases [Smyth & Cunningham, 1996]. In our implementation the system logs each time it needs to relax a query. When the same query term is encountered again the query is automatically relaxed by the previous amount. Whilst not a guaranteed improvement this heuristic, combined with the improvements in server side architecture, has reduced the server side retrieval time on average by almost 50%. Moreover, since the amount of query relaxation is controlled dynamically by the introspective learning heuristic it can reflect changes in the case-base due to the addition (or removal) of cases.

4 Conclusions

This implementation has shown how a distributed CBR system can be created on the web in a relatively short space of time (six months). Implementing the system for web delivery made the system much more viable. Just a few years ago we would have had to install the entire system (including the database of 10,000 records) on each of the sales staffs PCs. We would then have had to regularly send them updates to the database. This would have significantly increased the operational costs of the system. Thus the web is an ideal medium for delivering intelligent support of all types.

The project was most certainly helped by having a ready made case library. Although knowledge engineering work was still required in determining valid ways of relaxing the SQL queries and creating similarity metrics. This was not surprising as the similarity measure is one of the most important *knowledge containers* of any CBR system [Richter, 1998].

XML is a useful communications protocol enabling large packets of formatted information to be exchanged thereby reducing network traffic. As a possible replacement to HTML it should help the web support intelligent applications [Hayes, et al., 1998; Doyle, et al., 1998]. However, we have demonstrated that implementing a simple CBR system is not difficult and is a viable alternative to purchasing a commercial tool. We will see CBR systems playing an increasingly important role in product selection and specification on-line, since similarity-based retrieval is very useful for Internet e-commerce systems [Wilke, et al., 1998].

On another level this implementation confirms the growing realisation that CBR is a methodology for problem solving that can use a wide variety of approaches [Kamp et al., 1998; Watson, 1998]. Our system first uses SQL retrieval to obtain a set of broadly similar cases and then uses nearest neighbour retrieval to rank those cases. Further work is planned to investigate the possibility of providing support or possibly automating some aspects of case adaptation.

References

[Doyle, et al., 1998] Doyle, M., Ferrario, M.A, Hayes, C., Cunningham, P., Smyth, B. (1998). CBR Net: Smart Technology Over a Network, Internal Report Trinity College Dublin, TCD-CS-1998-07. http://www.cs.tcd.ie/Padraig.Cunningham/publications.html

[Hammond et al., 1996] Hammond, K.J., Burke, R., & Schmitt, K. (1996). A Case-Based Approach to Knowledge Navigation. In, Case-Based Reasoning: Experiences, Lessons, & Future Directions. Leake, D.B. (Ed.) pp.125-136. AAAI Press/The MIT Press Menlo Park, Calif., US.

[Hanney & Keane, 1996] Hanney, K. & Keane, M. (1996). Learning Adaptation Rules From a Case-Base. Advances in Case-Based Reasoning, Smith, I. & Faltings, B. (Eds.) Lecture Notes in AI # 1168 pp.179-192. Springer-Verlag, Berlin.

[Hayes, et al., 1998] Hayes, C., Doyle, M., Cunningham, P., (1998). Distributed CBR Using XML, Internal Report Trinity College Dublin, TCD--CS-1998-06. http://www.cs.tcd.ie/Padraig.Cunningham/publications.html

[Kamp et al., 1998] Kamp, G. Lange, S. & Globig, C. (1998). Case-Based Reasoning Technology: Related Areas. In, Case-Based Reasoning Technology: From Foundations to Application. Lenz, M. et al (Eds.) LNAI # 1400 pp.325-351. Springer-Verlag, Berlin.

[Kitano & Shimazu, 1996] Kitano, H., & Shimazu, H. (1996). The Experience Sharing Architecture: A Case Study in Corporate-Wide Case-Based Software Quality Control. In, Case-Based Reasoning: Experiences, Lessons, & Future Directions. Leake, D.B. (Ed.) pp.235-268. AAAI Press/The MIT Press Menlo Park,Calif., US.

[Leake, et al., 1995] Leake, D.B., Kinley, A. & Wilson, D. (1995). Learning to Improve Case Adaptation by Introspective Reasoning and CBR. In, Case-Based Reasoning Research & Development, Veloso, M. & Aamodt, A. (Eds.), Lecture Notes in AI # 1010, pp.229-240. Springer-Verlag, Berlin.

[Markovitch & Scott, 1993] Markovitch, S. & Scott, P.D. (1993). Information Filtering. Selection mechanisms in Learning Systems. Machine Learning, 10, pp.113-151.

[Richter, 1998] Richter, M. (1998). Introduction - the basic concepts of CBR. In, Case-Based Reasoning Technology: from foundations to applications. Lenz, M., Bartsch-Sporl, B., Burkhard. H-D. & Wess, S. (Eds.). Lecture Notes In AI # 1400 Springer-Verlag, Berlin.

[Shimazu, 1998] Shimazu, H. (1998). Textual Case-Based Reasoning System using XML on the World-Wide Web. To appear in the Proc. Of the 4th European Workshop on CBR (EWCBR98), Springer Verlag LNAI.

[Smyth & Cunningham, 1996]. Smyth, B., & Cunningham,). (1996). The Utility Problem Analysed: A Case-Based Reasoning Perspective. Advances in Case-Based Reasoning, Smith, I. & Faltings, B. (Eds.) Lecture Notes in AI # 1168 pp.392-399. Springer-Verlag, Berlin.

[Watson, 1997] Watson, I. (1997). Applying Case-Based Reasoning: techniques for enterprise systems. Morgan Kaufmann Publishers Inc. San Francisco, CA.

[Watson, 1998] Watson, I. (1998). Case-Based Reasoning is a Methodology not a Technology. Research & Development in Expert Systems XV, Mile, R., Moulton, M. & Bramer, M. (Eds.), pp.213-223. Springer-Verlag, London.

[Wilke, et al., 1998] Wilke, W. Lenz, M. Wess, S. (1998). Intelligent Sales Support with CBR. In, Case-Based Reasoning Technology: from foundations to applications. Lenz, M., Bartsch-Sporl, B., Burkhard. H-D. & Wess, S. (Eds.). Lecture Notes In AI # 1400 91-113. Springer-Verlag, Berlin.

[WWW Consortium 1997] World Wide Web Consortium, (1997). Extensible Markup Language 1.0, recommendation by W3C: www.w3.org/TR/PR-xml-971208

[WWW Consortium] World Wide Web Consortium home page: www.w3.org