Marir, F. & Watson, I. (1995). Can CBR imitate human intelligence and are such systems easy to design and maintain? In, Progress in Case-Based Reasoning, Watson, I. (Ed.), pp. 107-117, Springer Verlag, LNAI 1020, Berlin.

# Does CBR imitate human intelligence and are such systems easy to design, develop and maintain? A critique.

**Farhi Marir and Ian Watson**
*AI-CBR, University of Salford, Salford, M5 4WT, UK*
*www.ai-cbr.org*

## 1. Introduction

> "*Human intelligence is approximately ninety nine percent pattern recognition and one percent reasoning*" [Forsyth, 84].

This paper discusses the design and development expert systems that imitate human intelligence by solving discrete problems. This imitation has tended to focus on reasoning methods and the knowledge required to solve problems rather than focusing on the way problems are routinely solved. That is, reasoning rather than problem recognition. Case-based reasoning (CBR) is claimed to be a new paradigm that is more akin to the human way of solving problems. It has consequently been claimed that CBR systems will be easier to design and develop than model-based expert systems. This paper highlights some problems that exist with the model-based approach and asks if CBR really does address these problems. Our findings are based both on CBR literature and an investigation into the design of an expert system for housing refurbishment.

## 2. Evolution of Expert Systems

An important aim of AI is to imitate human intelligence in resolving, interpreting and explaining problems in real world domains. Designing expert systems requires building an explicit model of the knowledge needed to solve a problem. *Second generation* systems [Clancey, 85] use a *deep* causal model that enables a system to reason using first principles. But whether the knowledge is shallow or deep an explicit model of the domain must still be elicited and implemented often in the form of rules or perhaps more recently as object models.

Expert systems built using rules have been the predominant commercial applications of AI in the last decade [DTI, 92]. These systems contain hundreds of rules that represent a causal model of the problem domain. In a complex domain, particularly one that involves quantitative or experimental judgement, eliciting and encoding this knowledge into a rule-set can present great problem for the knowledge engineers. Moreover, such systems can be brittle [Hart, 85] and are difficult to modify or maintain as knowledge changes [Bachant & McDermot, 84; Coenen & Bench-Capon, 92; Watson et al., 92]. Moreover, they are often slow and are unable to access or manage large volumes of information [Gallaire 81, Minker 88, Marir 93].

Solutions to these problems have been sought through better elicitation techniques and tools [Motta et al., 89; Brooke & Jackson, 91], better KBS shells and environments, improved development methodologies [Diaper, 89; Inder & Filby, 91; Watson et al., 92a; Wielinga et al., 92], knowledge modelling languages and ontologies [Alexander et al., 86; Steels, 90; Chandraskaren 86 & 90; Wielinga et al., 92], and tools for maintaining systems [Bench-Capon & Coenen, 92, Watson et al., 92b].

Despite these efforts, several classes of potential applications, ranging from computer-aided design to medical applications, could not be implemented as they require efficient systems that have both the ability to manage large volumes of information and to perform deductions [Lai 92; Meyer et al. 91 and Kuhara et al. 91]. Attempts have been made to integrate database technologies with artificial intelligence (Gallaire 78, 81, Minker 88, Marir and Yip 92). In such integrated systems, the database management systems (DBMS) can be used more intelligently and efficiently if enhanced with logic system features such as the inference capabilities, while logic systems can effectively be made to access and share very large database through existing DBMS technology. The concurrency control, integrity constraint management and security enforcement of the database management systems will be very beneficial for the large and expensive knowledge bases that need to be shared and protected.

However, there is a more fundamental problem that has been overlooked. KBS practitioners did not consider how to build a KBS when there was no model available for complex domains. Overlooking this problem reflects the heritage of KBS in academic research laboratories. The early KBS (e.g., DENDRAL, MYCIN, PROSPECTOR) all operated in domains where there were good underlying models (either from first principles or statistical) - scientists are comfortable with working with models, they build them for a living. Unfortunately, in a commercial environment and outside of the Universities many people solve problems using previous experiences without reference to first principles and underlying causal or statistical models. It is no surprise that *expert* and *experience* derive from the same root. We posit that the KBS community was seduced by rules and neglected the truism that experts solve problems by applying their experience, whilst only novices attempt to solve problems by applying rules they have recently acquired. The application of experience to problem solving is the hallmark of CBR. CBR is proposed by some as a psychological theory of human cognition [Slade, 91] and one that provides a cognitive model of how people solve problems [Kolodner, 91]. It offers a paradigm that is claimed to be close to the way people solve problems and one that overcomes the brittleness of MBR systems [Barletta, 91; Helton, 91]. Also, CBR is attracting attention because it seems to directly addresses the problems faced in knowledge elicitation and maintenance outlined above, and integrate both database and AI techniques in storing, retrieving and reasoning upon the knowledge of the domain.

## 3. The CBRefurb Case-Based Expert System

In a complex domain, particularly one that involve qualitative and experiential judgement, eliciting and encoding this knowledge into a rule-set can present great problem for the knowledge engineers. That's one of the reason that we opted for case-based approach in the **CBRefurb** project . In addition, refurbishment represents a substantial part of the UK construction programme (by the year 2000 around 90% of the UK property will be over 20 years old) and refurbishment is seen as an economic alternative to new build. It reduces lost commercial and residential space and infrastructure expenses, and safeguards social relations and the architectural value of properties. Second, unlike new building, in this domain there are two interrelated source of complexity: the building's condition, which may not be accurately known until after a detailed building condition survey and the client's requirements, which may not be completely defined at the feasibility stage. An investigation amongst the North West of England Councils and contractors in this domain confirmed the facts mentioned above:

- this domain is characterised by nuances, variations, exceptions and too many interrelated factors, and
- the approach used for cost estimation of building refurbishment is based on previous experiences.

The **CBRefurb** project responds to this crucial need for the real world strategy of cost estimation by using previous experience as a base for estimating where uncertainty exists along with the use of estimation techniques and rules where appropriate. The design of a CBR system requires:

- an appropriate case representation of previously correctly solved problems,
- a definition of an explicit strategy to search for similar cases and
- define how cases can be adopted to match new situations.

**Case Representation**

At least three different types of factors have been identified as having an impact on the refurbishment price:

- The building specification features which describe the type of refurbishment work, the occupancy, the site access, and the state, the type and the age of the house, etc. Only experience can tell when these feature may or may not impact on the price of the work.
- The second factors are external and they reflect the state of the market, technological innovations of new and cheaper materials, the bidding season and the personal experience of the estimator.
- Finally, the item specific features which describe the type, size, the current state, the type of repair required and the quality of materials used for this repair for each item that have been broken down in the specification document. These features have a direct impact on the price of refurbishing building and they can be sometime explicitly calculated using appropriate estimation techniques.

It results from this that a refurbishment case will be composed of three parts:

- The first part contains the building specification such as the type, the age and the state of house, the type and the quality of the work required such as complete or partial refurbishment, the real costs for the whole work and finally advice and problems that emerge during work and the way they have been solved.
- The second part highlights the external factors that impact on the price of the work at the time of refurbishment like for example the aim of the work and the state of the market.

- The third part of the case is a set of sub-cases each of which reflects the information on the work for each specific item of the building e.g. foundation, walls and the roof. Each sub-case contains the item specific features which describe the type, the size, the current state of the item, and the type and quality of repair required, the real cost of the item work, advice or problems met during the work, and the guarantees and conditions applied to the work.

A hierarchical organisation of these features as indexes is retained. The more important features that describes the house to refurbish and the client requirements are at the top of this hierarchy representation. Then comes the sub-cases specific features. This organisation preserves the existence of generalised and specific features in each path of the hierarchy.

## Case Retrieval

Most CBR systems use a single 'best' or 'most similar' case as the basis for their solution [DARPA 91, Kolodner 93], but in the refurbishment domain, the solution to a new problem involves pieces of several old cases. Thus a strategy of multiple features and multiple case retrieval is more appropriate. In this respect, the house specification features and external features to which weight can be associated by the user, are used as generalised features. Using the nearest neighbour technique [Cognitive Systems 92] and refurbishment domain dependent knowledge, cases that respond to these generalised features are selected. These first selected cases are called ***Context-Cases*** and are recognised as members of a top level class. Once these context-cases are selected further refinement can be performed to choose more useful cases among them, using the refurbishment domain knowledge. Then mixing nearest neighbour, induction techniques [Quinlan 79] and item specific domain knowledge, the specific features of each sub-cases are used to guide search downward through the hierarchy until a set of similar source sub-cases are reached. In the event where no sub-cases match the current sub-case or some of its important features, backtracking up the hierarchy may be used to return to previous selected cases and perform an extensive search for cases that match the remaining unmatched features.

## Case Adaptation

It is rare to find a *perfect* previous case that matches *exactly* the new case. For this reason a case adaptation process will be implemented using a combination of procedural programming, rule-based and CBR techniques. In this system, the adaptation process starts at the level of the item sub-cases. If the item is similar then readjustment of the price can be performed. However if the sub-case does not exactly match or no similar sub-cases exist, different techniques such as cost functional unit, square meter method, approximate quantity or a source of reference [Smith 92] can be used to compute the price of the item. Adding to that, any house specification, external and interrelation between items factors which have an impact on the item, will be taken in the account during this item sub-cases cost adaptation. The outcome from this sub-cases adaptation process will be not only the estimated price of the item but also any advice on the strategy of the work, the material to use or any problems to prevent.

The same approach will be used for each item sub-case and extended to the whole case.

## 4. A critique of CBR

We found during the investigation of cost estimation in building refurbishment that most of the estimators in the North West of England use previous case estimation as a basis for any new refurbishment work. They also confirm that they use other estimation techniques [Ref] only when previous similar cases are not existent, and always these techniques give results far from reality in contrast to the close estimations to the reality when using previous real cases.

To this end, this section presents a comparison between model-based and case-based reasoning, from the aspect of the imitation of human expertise and the practical design of expert systems.

### 4.1 Imitating human expertise

Expert systems are designed to imitate human expertise; they can solve problems and explaining their reasoning by explicitely representing and managing large amount of complex knowledge.

**Problem Solving**

Hoever, People do not solve problems using conceptual models of the world and the knowledge of an expert can not be entirely embodied in a set of rules or various hypothetical models. People do not reason about each problem they face from first principles as if they nor anyone else had ever faced a problem like it before. Instead, they try to find the best plan they have heard of, or previously used, that is the closest to the problem at hand and attempt to adapt the plan to the current situations (Riesbeck & Schank 1989). These are the reason why much of the AI community agrees that CBR is more close to the way people solve problems and thus as a psychological theory of human cognition [Slade, 91; Gonzalez & Laupeano-Ortiz 1992] and one that provides a cognitive model of how people solve problems [Kolodner, 91].

Comparing case-based reasoning to model-based reasoning from the human imitation aspect, is in fact comparing the use of experience and models as a way of solving problems. CBR is more efficient than model-based since reasoning from first principles is often a very complex way to come to an answer. Many steps must be followed through and often many assumptions must be made and checked. If human expertise was based on this method then everyday tasks would require a huge intellectual effort. However, using past experience is often a quicker and simpler method of reaching a solution. An illustrative example of the importance of previous experience is found in some councils who often refurbish a single house and use the resulting experience as an estimate for refurbishing large schemes of similar houses.

However, claims by the CBR community should not deny that model-based reasoning can provide accurate and precise results for well formulated problems as compared to the often approximate results of CBR. The techniques of CBR can be made more accurate only if it integrates MBR into the techniques of retrieval and adaptation [Ref]. We found that even in weak domains, like refurbishment, parts of these domains can be modelled and implemented using rules and object-oriented methods alongside CBR techniques.

## Explanation

Many questions arise whether most current model-based systems can really claim to be *experts* in their domains. One of the main areas of criticism is the quality and depth of the explanation of their reasoning in complex domains. A main cause of inadequate explanation is that a system while containing sufficient knowledge to infer a useful answer, lacks the underlying expertise upon which the performance is based [Ref]. A human expert can explain his behaviour, not merely retrace his steps, by motivating and validating his decision and relating them to the domain as necessary. However, people generally prefer to reason from past experience rather than from theoretical knoweledge because solutions erived from past experiences have precedents. Thus, answers given by case-based system can be explained and justified with precednets and not by listing the rules that fired or summarising the principals used to arrive at the answer. This is supported by our study. Many refurbishment experts found it easy to explain and convince their executive managers using real cases when making decisions ranging from minor ones (e.g., using one material instead of another) to strategic decisions (e.g., deciding between redeveloping or refurbishing a scheme).

## Managing large and complex knowledge

Many expert systems focus on inferencing and can not manage large volumes of complex knowledge (Gallaire 1978, 1981, Minker 1988, Marir and Yip 1992). Due to such limitations, several classes of potential applications, ranging from computer-aided design to medical applications, can not be implemented since they require efficient systems that have both the ability to manage large volume of information and to perform deductions [Lai 1992; Meyer et al. 1991 and Kuhara et al. 1991]. However, in integrated database and AI systems, the database management systems (DBMS) can be used more intelligently and efficiently if enhanced with logic system features such as inference capabilities. While logic systems can effectively be made to access and share very large database through existing DBMS technology. The concurrency control, integrity constraint management and security enforcement of the database management systems will be very beneficial for the large and expensive knowledge bases that need to be shared and protected. The ability of an integrated system to reasons and manage large and complex knowledge-bases is claimed to be a characteristic of CBR. However, there is little research that shows that CBR systems can manage and reason with massive case-bases. The *scaleability* of CBR is an pressing research issue.

## Learning Process

It might seems from the literature that the learning process in CBR involves simply adding new cases to the case base. However we believe that the process is not so simple and must involve the following processes:

- Check the validity of the case instead of simply adding it, since incorrect cases may cause system to work improperly and redundant cases may cause inefficiency Relying on previous experience without validation may result in inefficient or incorrect solutions being recommended causing an increase in problem-solving time or errors that may have negative effects on the process of learning.
- Forget unused cases in order to maintain case-base efficiency.
- learn about the emergence of any indices that had not previously been thought significant. It is very important to have a dynamic mechanism for indexing features where some features who used to be meaningless can be very important. For example, in construction some buildings may have to demolished instead of refurbished because the cost of meeting modern building standards would be prohibitively high.

## 4.2 Design, Development & Operation

Although expert systems, especially rule-based and object-based are popular, they are, however, still regarded by many as an expensive luxury and they still face several problems during their design, development and operation. Namely:

- knowledge elicitation is a difficult process, often being referred to as the *knowledge elicitation bottleneck*;
- implementing KBS is a difficult process requiring special skills and often taking many man years;
- once implemented model-based KBS are often slow and are unable to access or manage large volumes of information; and
- once implemented they are difficult to maintain [Bachant & McDermot, 84; Coenen & Bench-Capon, 92; Watson et al., 92b].

Facing these problems the AI community is attracted to CBR because it claims to directly address the problems outlined above:

- CBR does not require an explicit domain model and so elicitation becomes a simple task of gathering case histories,
- implementation is reduced to identifying significant features that describe a case, an easier task than creating an explicit model,
- by applying database techniques largely volumes of information can be managed, and
- CBR systems can learn by acquiring new knowledge as cases thus making maintenance easier.

### Knowledge elicitation and acquisition

The process of elicitation and acquiring knowledge usually require both a domain expert and a knowledge engineer. A large amount of time is needed to obtain and process the knowledge required for any reasonable sized domain. This stage represents an expensive *knowledge elicitation bottleneck* [Hayes-Roth et al., 1983]. Moreover, the engineer will meet a lot of problems in acquiring knowledge even in a domain that is based on a strong theory.

The fact that CBR systems use knowledge in a form familiar to the expert makes knowledge transfer between the domain expert and the ystem simpler. In domains that already have much of the required knowledge in the form of cases, CBR systems are easy to develop. Prior examples are compiled, analysed and input into a case base, instead of having to acquire and represent knowledge in the form of rules [Yoon et al., 93]. CBR is expected to overcome the knowledge acquisition problem [Kobayshi 92]. However, it should be mentioned that CBR does not necessarily remove the need for knowledge acquisition altogether. What is being suggested is that there could be a major saving in the amount of knowledge engineering required to produce a CBR system compared to a MBR systems [Hennessey et al., 92].

Although, there may already be existing cases, choosing the attributes that are used to describe cases may require specialised knowledge engineering skills [Dearden et al., 93]. We face this problem in the refurbishment domain where the records we collected are in standard format but the emphasise on the features is very different from one enterprise to another. Moreover, If MBR faced the problem in eliciting and acquiring knowledge to build their models, case-based systems will face similar problems when using rules or formulae for adaptation. Since some items in our refurbishment domain will be modelled and presented using rules or objects. The elicitation and acquisition of this domain knowledge will obviously meet the same elicitation problems as faced in model-based systems.

**Expert system development**

A CBR application can help the developer get the application running quickly, even though there is an incomplete case library [Hennessey, 92]. An incomplete rule-based system provides little value. This is because rule-based systems matche rules to a problem description, a missing rule will halt the reasoning process. The problem will not be solved. However, partial matching and “best guessing” are built into the case-based strategy, because it is seldom that two complex situations match completely [Slator et al., 92]. Other nearly matching cases can compensate for a missing case. Therefore, the system can find and adapt at least a partial solution. However, difficulties may still arise regarding the confidence in the data collected for individual cases [Dearden et al., 93]. Moreover, Inference, who have considerable expereince in fielding case-bases have reported that if a system can not solve most problems adequately there can be a very negative responce from users. This can jeopardise the success of the system.

The claim that CBR systems can be implemented faster than MBR systems was supported by a study conducted by Cognitive Systems which stated that it took two weeks to develop a case-based version of a system that took four months to build in rule-based form [Goodman 89]. Also, and more recently, developers at Digital Equipment Corporation confirmed that a rule-based system called CANASTA took more than eight times longer to develop than CASCADE a case-based system with the same functionality [Simoudis, 92; Simoudis et al., 93]. However, claims such as these should be treated with cautions. The fact that the knowledge acquisition and elicitation have been performed when first developing these systems using rules, may contribute to the speed of developing them using CBR.

### Maintenance

By their very nature expert systems require regular updates and maintenance. In all but the most static domains knowledge is continually changing. This need for regular maintenance is in addition to the general requirement to debug or expand computer systems in general. For many years practitioners believed that expert systems were easy to maintain - almost all books on expert systems development written during the eighties will contain a quote similar to “maintaining a rule-base is easy, being simply a matter of adding or subtracting rules from the knowledge-base” Easier than maintaining procedural C or FORTRAN code true, but not *easy*. Unfortunately, the experience of XCON/R1 [Bachant & McDermot, 84] and others [Coenen, & Bench-Capon, 92; Vargas & Raj, 93] has shown that maintaining model-based systems is not as simple as adding or subtracted rules or objects. As a knowledge-base grows it becomes a complex debugging task. The maintenance of the knowledge-base in expert systems usually requires the re-employment of a knowledge engineer and in larger systems the splitting of control knowledge and data can hamper maintenance. Introducing a new rule or modifying an existing one where the rule-base is large may lead to clashes of rules. There are no widely accepted techniques or procedures for maintenance of these systems. However, cases are easier to maintain [Slator, 92] since there is no requirement to edit a rule set or construct a decision tree; the system easily absorbs new experiences. Indeed a CBR system can grow as it gains experience of more cases.

Another major benefit claimed for CBR is a case-base may be updated and modified by the expert without the assistance of a knowledge engineer or developer. The effort of maintaining the knowledge base and the consequent cost and time are therefore greatly reduced. However, such *user-maintenance* has only been reported with Inference’s tool CBR Express.

The above claims for CBR are still to be proven. As mentioned above it is unwise to blindly add cases to a case-base. Moreover, if developers have integrated rules into their CBR systems for adaptation, there will be a need to maintain the rules.

## 4 Conclusion and future work

The investigation and the prototyping of **CBRefurb** have been useful in acquiring experience of theoretical and practical aspects of case-based reasoning. This paper critiques the claims made for CBR. As with many claims there is often some truth. CBR does seem to mimic human reasoning. CBR systems do seem to ease the knowledge elicitation bottleneck, they may be easier and quicker to implement and they can learn. However, our work shows that developers of CBR systems do face real problems. It is necessary to integrate CBR fully with other reasoning paradigms and information systems. Methods for analysing and maintaining cases need formulating and importantly the effectiveness of CBR in commercial applications must be critically evaluated.

## Acknowledgements

This work was partially funded by EPSRC project number GR/J42496.

## References

CIOB (Chartred Institute of building) [1987]: Refurbishment and Modernisation.

Cedric Pugh [1990]. The costs and benefits of rehabilitation and refurbishment Proc. of Int. Symposium on Property Maintenance, management and modernisation, 7-9 March 1990, Singapore, Ed. Quah, Lee Kiang.

Douglas J.F. and Peter S.B. (1991). Cost Planning of Buildings (6th Edition). BSP Professional Books.

Summers, A.P. and Fellows, R.F. (1987). How to refurbish occupied buildings. Building Technology and Management, Aug/Sept 1987, pp34-5.

Todd S. (1991). Refurbishment of the Denis-Wild House'. Managing, Quality and Economics in Building, Artur Bezelga and Peter Brandon (Ed.). Publisher E & FN Spon.