The International Journal of Digital Accounting Research
Vol. 1, No 2, pp. 111-123
ISSN: 1577-8517

Issues for Auditors Designing Case-Based Reasoning Systems

Olivier Curet. City University and Deloitte & Touche. UK.

Mary Jackson. London Business School. UK.

Abstract. This paper examines the main methodological issues to be considered for case-based reasoning (CBR) systems. The advantages of knowledge representation in cases are discussed, giving the rationale for these systems. Many different aspects of design are considered, including user requirements where the system is intended to encourage user learning. A framework for designing such case-based learning and reasoning (CB-LR) systems is discussed. The focus is on feature calibration and case stabilisation processes, together with issues concerning implementation and evaluation of systems.

Key words: *artificial intelligence, case-based reasoning, case-based learning and reasoning.*

1. THE ADVANTAGES OF KNOWLEDGE REPRESENTATION IN CASES: INTRODUCTION

To understand some of the advantages of knowledge representation in cases, it helps to contrast ‘rule’ representation (mainly used for traditional rule-based expert systems) and ‘case’ representation (mainly used for CBR models). There are four major advantages with respect to cases: the legitimacy of the knowledge representation paradigm, the evolving coverage of the problem domain (which helps to solve the maintenance problem), the proximity of the use of cases to human learning and reasoning, and the relevance of the case approach for experience-rich domains which lack theories.

First, cases may be more legitimate as knowledge support and less prescriptive than conclusions reached from rule processing. Because the knowledge is in case form, the cases retrieved generate knowledge which may be more meaningful to the user: it has been shown that when faced with a difficult problem, the expert will “often look to analogous problems for possible solutions” (Lunce et al 1993).

The case form may also be more meaningful since experts can explore the information available in the case for signs and hints that could shorten the logical sequence used to reach a decision. At the same time, the experts who contribute cases (either in the design or use phase) may advance their own understanding of the problem domain. Rather than debating the correctness of the rule set, they are able to benchmark their own cases when pooled together with others.

Second, the coverage of the problem domain extends naturally. Rule representation has sometimes resulted in circular reasoning, conflicting rules, or dead-end conclusions. Doubts have been cast on the ability of rules to reflect the completeness of structured knowledge. In changing problem domains, it is difficult to ensure that rules sets are up-to-date, encapsulating any recent changes. However, cases can be stored as they occur and later processed in many different ways.

The case base evolves with time, and shifts in the experts’ views and solutions strategies can be accommodated dynamically. Although the accumulation of cases can be a natural and unconstrained process, case elicitation must be managed carefully. Recent methods such as the calibration of reference case libraries have been reported (see Curet et al 1995).

Third, reasoning from cases may come closer to human cognitive processes. There is evidence that people naturally use cases in their normal decision-making: “People learning a new skill often refer back to previous problems to refresh their memories on how to do the task” (Kolodner 1993). It has also been stated that “experts think more holistically, need fewer cues, and rely on analogs” (Silverman 1992). It has also been suggested that human learning is dependent on the accumulation of past cases: “learning is enhanced through having realistic experiences (...) The key is in being able to learn from experiences as they are gained and to put concepts, models and theories into context and into practice on a continual basis” (Angehrn et al 1993). For example, diagnostic skills are acquired mainly by experiencing the act of diagnosis itself (Boreham 1987). In specific situations, humans may prefer to relate to cases rather than models, since it is easier to think of a case having been witnessed and then to describe it in its context rather than trying to formalise it with mere rule representations.

Fourth, the relevance of the case approach for experience-rich domains which lack theories. Case representation is an appropriate format in experience-rich domains which lack theories, where not all the highly-skilled knowledge can be formalised. For example, when it comes to managerial fraud there is no general theory of what constitutes an examplar fraud case (Curet et al 1995). Even if there was a general theory of what constitutes a prototypical instance of a management fraud case, this would be invalid as new economic, financial and even sociological patterns emerge.

Case representation has been shown to be valid in a range of domains. Ideally, the problem domain should have the following characteristics:

- unstructured (if the problem is relatively well structured it may lend itself to more procedural formalisms),
- inconsistent (expertise exists collectively but is difficult to comprehend since there is no interpretative framework for the problem domain),
- dynamic (the domain changes over time and different patterns emerge due to the changes),
- precedent-based (the knowledge is restricted to cases mainly, i.e. the law domain for which the role of case adaptation is vital),
- dispersed (over time or geography),
- scarce (cases do not happen frequently).

As software tools improve and successful applications proliferate, the characteristics of suitable problem domains are becoming more clearly delineated.

2. ISSUES IN DESIGNING CBR SYSTEMS

From experience gained from providing systems to assist auditors in specific areas of decision making (Curet 1995), the following issues arise in designing CBR applications.

2.1. Initial Problems due to the Nature of CBR Systems

Designing CBR systems has been problematic and controversial for several reasons:

- as CBR is relatively new, there is no fully-fledged, well-accepted general methodology to build such systems. However according to Watson (1994), expert systems practitioners "did not consider how to build an expert system when there was no model available. Overlooking this problem reflects the heritage of expert systems in academic research laboratories". Business need usually drives the derivation of a methodology.

- although a CBR system encapsulates knowledge, the system design and methodology for building differs crucially from that appropriate for traditional expert systems. Past methodologies for developing rule-based expert systems (RBES) cannot be adapted to suit CBR systems: solutions, proposed on the basis of retrieved cases with adaptation depend on the number and "quality" of cases, the adaptation processes, and the evolutionary nature of the knowledge modelled requires controlled case management.
- appropriate design methodologies depend on the purpose of the system and its desired outcome(s). Recently the classification of case-based systems has been discussed in the literature, and Figure 1 illustrates one approach (Bradley et al 1995). This classification differentiates systems on their case "representation complexity" (including issues such as case acquisition and domain completeness) and their "similarity complexity" (i.e. where the similarity levels between the current case and the past ones lie on the spectrum from 'straightforward' to 'difficult'). The design approach is likely to differ for each quadrant in the classification diagram:

REPRESENTATION COMPLEXITY	II ξξξ - structured cases Unambiguous indexes	IV ξξξ - structured cases Ambiguous indexes
	I Highly structured cases Unambiguous indexes	III Highly structured cases Ambiguous indexes

SIMILARITY COMPLEXITY

Figure 1 A Framework for CBR Systems

2.2 System-Centred Issues

The following paragraphs highlight some of the important system-centred issues.

- selecting CBR software for developing applications depends on the objectives of the application. The selection of the software is not straightforward: the question must be raised about whether it is more appropriate to acquire a CBR shell or customise a CBR application with in-house resources.
- a central component of case-based reasoners is their various case retrieval methods. All CBR shells use the nearest neighbour approach to identify similar

cases. In addition, some CBR shells use induction algorithms (such as ID3 or CART) to separate cases into sets of similar cases. However, induction algorithms also have a few limitations (Aha 1991) such as the fact that some of the information requested for cases may be unnecessary.

To remedy this, the possibility of combining induction and rule-based search is presently being investigated (Kamalendu et al 1996). Alternative inductive algorithms such as IBID are being developed especially for integrating machine learning, problem solving and explanation. Whatever inductive algorithm is being used, it is necessary to consider which type of search is appropriate for the particular use and problem domain.

- CBR systems emphasise analogical problem solving, that is, problem solving by imitation (i.e. the user applies solutions to the current problem by referring to past solved cases). Analogical reasoning provides an opportunity for the user to justify and support his or her decision when the domain is complex or when there is a need for conflict resolution.
- some CBR systems explicitly aim to combine search with learning. Using CBR may give the user access to deeper knowledge and more relevant reasoning about the problem in the form of a ‘data laboratory exercise’. In contrast to the ‘result-orientation’ of RBES, the searching and learning CBR approach can have a ‘critic orientation’. A few recent research papers have emphasised the use of CBR to improve the skills of less experienced personnel, in engineering fields (Rudiger 1994), as well as in finance (Johnson 1995).

The CBR application can be used either as a ‘directing’ system (using cases to provide the user with simple adapted solutions from past cases relevant to the case under scrutiny) or as an ‘indicating’ system (giving the user an opportunity to discover knowledge from cases which are ‘neighbouring’ the problem).

2.3 User-Centred Issues

The main user-centred issues are as follows:

- psychological studies have shown that people can take only a few issues into account when making decisions. Decision-making has a tendency to be conventional and to ‘regress towards the mean’. Decisions tend to be tradition-bound, resulting from shallow investigation, and to be uninfluenced by new information. When a decision is made, it is often the result of applying standard procedures and will adjust only slowly to changing conditions.

CBR can be used to remind decision makers of many specific factors which were considered relevant in past cases which may now have been forgotten. If the combination of human decision maker and computer leads to better decisions, the evolving human-machine relationship may help reduce users' reluctance to engage with CBR.

- Information retrieval with a case-based system is best presented visually, with graphical representations of cases and retrieved knowledge. For example, the branches in a decision tree are clearer in a graphic representation than a string of conditions. Similarly, the shapes and density of clusters provide information on clustering structure with most impact if presented visually (Frawley et al 1992).
- it is worth pointing out that a heightened level of interaction is required to learn new skills or solution strategies with a case-based system. An analogy is with data mining, which relies very much on data visualisation, viewpoints from different perspectives, segmentation into clusters on different features, and the application of additional machine learning techniques such as neural nets and rule induction.

Data visualisation means looking at data in many different ways, especially graphical representations. The process is exploratory and interactive. With this type of use, it is possible to learn from the system's feedback so that further investigations can take place. In a similar way, CBR involves searching, evaluating, trying to solve problems and then searching again iteratively, with knowledge gained from success (or failure) can be fed back into the system (Rowe 1993). The interactive exploration required in CBR can be viewed as a 'case-driven bottom-up' search. In considering the process of case exploration, issues such as accuracy and relevance (of the retrieved cases with respect to the user's expectations) or efficiency (time required from the user for the search process) need to be considered.

- Silverman (1992) mentioned several factors for successful human-computer collaboration for systems which critique human error. The first factor is the "mutual and continuous adaptivity", the second "remembering and analogizing". The first factor aims to ensure that the user can "grow and learn" from experience: "in the man-machine collaboration this implies the human should grow and learn about the domain from the strategies his problem solving behaviour precipitates. The machine, in turn, should become more knowledgeable the more it is used". The second factor is concerned with the level of success in collaboration which appears to expand "the more experience the collaborators can bring to bear".

To summarise, the user-centred as well as systems-centred issues need to be resolved if effective case-based systems are to be developed. Visual representations of cases and their associate findings displayed in a decision tree form enhance decision making. In some domains, (e.g. novice auditors improving their decision making with respect to fraud), case-based reasoning can be applied to support learning which in turn, supports reasoning. In these instances, case-based reasoning systems is best considered as case-based learning and reasoning (CB-LR), which forms a natural extension to straightforward CBR applications.

3. A METHODOLOGY FOR DESIGNING AND EVALUATING CB-LR SYSTEMS

The challenge for CB-LR is therefore to find methodologies for designing case-based systems to enhance the understanding and learning aspects for users (Aamodt 1994). Knowledge engineering (KE) translates into the systematic collection of a range of most relevant cases. After case collection, come eliciting the right knowledge, modelling it, testing processes and evaluating outcomes (relying also on people's expectations). The main issues for CB-LR as well as CBR design and evaluation revolve around the cases (their use and their appropriateness), the system itself (both building and validity), and the user's perspective (the impact of the query process).

Table 1 shows the main stages for CB-LR design, implementation and evaluation together with some suggested methods. These stages were undertaken in a specific application in audit (Curet and Jackson 1995) and they are reminiscent of the stages in early RBES rapid proto-typing. It is likely that they will be appropriate in other business domains where decision-making is precedent-based and for which previous cases need to be recorded and recalled. The stages are:

<u>Feature calibration process</u>: the first part of the methodology consists of case feature definition. 'Features' is the term used in this paper for the case descriptors. One method for constructing a set of potential case descriptors (used successfully in the audit application) consists of circulating a questionnaire which collects features from experts. Initially, the features suggested should arise from past cases, which allows experts who witnessed these cases to input their knowledge in a less constrained way.

Stage	Issue	Methodology
Case features definition	• case feature definition • coverage of problem domain	• semi-Delphic process • feature calibration process
Case base build-up	• case aggregation • appropriateness of case base	• case stabilisation process • on-going testing CBase grows
Use of CB-LR system	• examine of retrieval processes • query and adaptation process	• importance weights setting • user-guided weights possible
Evaluation	• “white box” or “black box” • impact on task, person & firm	• verification and validation • on-going approach

Table 1 Stages in CB-LR and suggested methods

The resulting set of features should be re-circulated to permit the experts to change, amend or delete any features they feel are inappropriate and the process repeated until the different experts agree. The amended form is circulated again to all the experts who crafted the questions in the first place, until the final version has been agreed (validated). This ‘semi-Delphic’ process allows the users and designers to agree on both the features that should be used to characterise cases and also to decide the types of cases that should be collected.

An alternative approach was designed by Allen (1994) who used source material at hand to “seed” a case-based library. Allen’s process is an alternative as far as the collection of primary data is concerned. From time to time, the feature set should be presented to the panel of selected experts so that they can be appropriately debated and developed. ‘Feature calibration’ is the process by which the set of case features changes with time, since the experts are asked to add, amend or delete any previous features.

The case base stabilisation process: after the feature calibration has been validated, it may be necessary to collect further cases on the basis of the agreed set. The purpose of ‘case stabilisation’ is to collect sufficient cases to obtain an appropriate coverage of the problem domain. Issues such as the effects of case aggregation (e.g. is there a target number of cases to collect) and case duplication (e.g. what should be done about redundant cases) must be tackled. Some empirical findings on the issue of aggregation suggest that a minimum of 50 cases per possible outcome are required, but the number also depends on the domain, the criteria being used, and the types of data in the cases. This so-called ‘shaping up’ process establishes the ‘reference’ case base, also called the case library.

Thereafter, the reference case base will evolve with use over time. On-going testing of the case-base should be conducted as it is ‘shaped up’. One approach is to evaluate the case base as each tenth of the target number of cases is reached. If need be, fictitious cases can be input during the ‘shaping up’ and if necessary features can be redefined. The overall purpose of both feature calibration and case stabilisation is to minimise the prototyping risk related to knowledge engineering: for too long, knowledge engineering has been the main risk factor in the knowledge-based systems design and implementation (Gammack et al 1985).

Figure 2 shows the flowchart which at the lower half consists of the steps (feature calibration and case stabilisation) and the upper half consists of the use and consultation of the system. suggested for designing CB-LR applications. The flowchart has been derived from Schank and Riesbeck’s work (1989).

Implementation process: once the case base has been stabilised (or once the target number of cases has been reached), it becomes the case library and the system is ready for customisation to the environment in which it will be used. The most appropriate method of case retrieval must be decided - inductive retrieval or nearest neighbour matching. Usually, this will include deciding the relative importance (or weights) of features in case retrievals and whether weight vectors should be prescribed or left open for users to choose. The flexibility of querying the case library has also to be examined: for example, to what extent does retrieving and creating vectors assist user in learning about the domain?, to what extent should the system rely on the user to define weighting and vectors?. In normal use, new (or hypothetical) cases to be solved are input by the user, examined by the system and solutions adapted or suggested on the basis of matching cases. If there is no perfect match for the case currently under investigation, then the most similar case(s) can be adapted to the present problem.

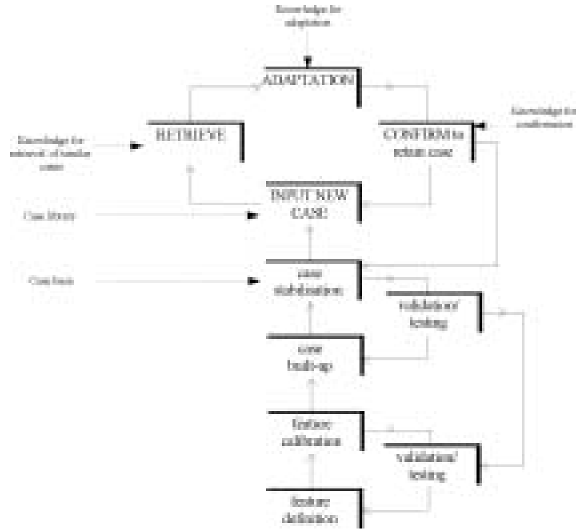

Figure 2 Flowchart for CB-LR (after Schank & Riesbeck)

The new case, for which a solution has been found may be confirmed and stored so that the system learns. If the new case raises a new point that the system has not encountered before, there may be a need to go through the case stabilisation process again, taking the new point into consideration. The adaptation process (by which solutions to similar cases are adapted) can be either system-guided or user-guided. If it is user-guided, the user has to scan through the most similar cases to support his judgement, not merely replace it. Aamodt (1994) has defined this as a combination of exemplar-based reasoning (i.e. the outcome of the most similar past case in the library becomes the solution to the present problem, in which case no adaptation is needed) together with memory-based reasoning (i.e. the reasoning results from the process of accessing and searching in the case library).

However, the exact purpose of the system leads to specific questions, for example: when should a new case be stored and by whom; who is responsible for ensuring that the system has been 'trained'; what kind of training is required (should it be technical or simply task-specific) and who should be responsible for the on-going evaluation of the system. For a case-based system that is used in geographically dispersed locations, it may be necessary to have an operator (who is responsible for inputting the data) and an evaluator (who is responsible for assessing the output).

The evaluation process: As already stressed, evaluation is best carried out in an on-going manner, as the system becomes better accepted and used Three alternative approaches may be possible: verification (or 'white box' evaluation), validation (or 'black box' evaluation), or a combination of the two. The white box approach is more concerned with the efficiency and the internal processes of the system (Terano 1994). This includes the measurement of the time and cost aspects associated with using the system (for example the time necessary for retrieval and adaptation processes), and the measurement of improvements in the case library achieved by expansion.

The verification process may include the use of case sampling for testing. For example, cases for which the outcomes are known can be chosen at random from the case base, run through the system and the results compared to their actual outcome. The 'black box' approach is more concerned with the user's perception of the system, and the impact of the system on the organisation. in evaluating case-based systems However, a different stance on evaluating CBR systems, takes into account not only internal issues relating to the structure (i.e. accuracy of the system), but also external factors (such as user acceptance) or behaviour (Althoff

1996). This more holistic approach was adopted in the cited audit application (Curet et al 1996) .

4. CONCLUSION

The experience of designing a methodology for CB-LR systems permits the following conclusion: the implementation process has to provide a way for experts to agree on case descriptions so that the application may encompass an appropriate coverage of the problem domain. The feature calibration process and feature stabilisation explained in this paper have contributed to doing so. The use of CB-LR should put the emphasis on searching, solving and learning issues: it is not enough for the user to have to access to cases that have specific similarities with the present situation. Users should be able to understand the value of navigating through individual cases or clusters of cases in a non-linear manner to support learning. CBR and CB-LR systems should be understood as paradigms by which users have access to a creative representation of knowledge and through which problems can be contextualised.

5. REFERENCES

Aamodt, A., and Plaza, E. (1994): “Case-Based Reasoning: foundational issues, methodological variations, and systems approaches”, *AI Communications*, vol.7, n.1.

Aha, D.W., Kibler, D., and Albert, M.K. (1991): “Instance-based learning algorithms”, *Machine Learning*, n.6:37-66.

Allen, B.P. (1994): “Case-based reasoning: business applications”, *Communications of the ACM*, vol.37, n.3:40-42.

Althoff, K.D., and Weis, K.H. (1996): “An evaluation of the INRECA CBR system for decision support and diagnosis”, paper presented at the 4th German Workshop on CBR - system development and evaluation.

Angehrn, A., and Doz, Y. (1993): “Business Navigator: The next generation of management development tools”, ICEDR White Paper.

Boreham, N.C. (1987): “Learning from experience in diagnostic problem solving”, *Student learning: Research in education and cognitive psychology*, Richardson et al (Ed.), Open University:89-97.

Bradley, J., and Gupta, U. (1995): “A classification framework for case-based reasoning systems”, *The New Review of Applied Expert Systems,*: 35-48.

Curet, O. (1995): “The application of case based reasoning to assist accountants in identifying top managemnet fraud”, *Proceedings of the Case Based Reasoning: Prospects for Applications*, Institute of Electrical Engineers.

Curet O., and Jackson, M. (1995): “Tackling cognitive biases in the detection of top management fraud with the use of case-based reasoning”, *Applications and Innovations in Expert Systems*, BCS series.

Curet, O., and Jackson, M. (1997): “A Case-Based Learning and Reasoning Agent in Unstructured Decision Making”, Paper presented at the III International Meeting on AI in Accounting, Finance and Tax, Huelva

Frawley, W.J., Piatetsky-Shapiro, G., and Matheus, C.J. (1992): “Knowledge discovery in databases: an overview”, *AI magazine,*: 57-70.

Gammack, J.G., and Young, R.M. (1985): “Psychological techniques for eliciting expert knowledge”, *Research and Development in expert systems*, Cambridge University Press: 105-112.

Johnson, P. (1995): “Selecting, extracting, and preparing data with machine learning”, *Artificial Intelligence in Capital Markets*, Probus: 595-623.

Kamalendu, P., and Campbell, J.A. (1996): “Integrating rule-based and case-based reasoning in legal decision-making processes”, Paper presentad at the 4th German Workshop on CBR - system development and evaluation.

Kolodner, J.L. (1993): *Case Based Reasoning*, Morgan Kaufman.

Lunce, S.E., Iyer, R.K., Courtney, L.M., and Schkade, L.L. (1993): “Experts and expertise: An identification paradox.”, *Industrial Management and Data Systems*, vol.93, n.9:3-9.

Riesbeck, C.K. (1988): “An interface for case-based knowledge acquisition”, Darpa Workshop Proceedings on Case-Based Reasoning: 312-326.

Rowe, J.A. (1993): “The meta logic of cognitively based heuristics”, *Expert Systems in Business and Finance, Issues and Applications*, Wiley,:109-127.

Rudiger, B., Henderson, W., Wrigley, D., and Wilson, J. (1994): “A case-based reasoning system for troubleshooting”, *Proceedings of Case-Based Reasoning: Prospects for Applications*, Institution of Electrical Engineers.

Silverman, B.G. (1992): *Critiquing human error - a knowledge-based human-computer collaboration approach*, Academic Press, San Diego.

Watson, I.D., and Abdullah, S. (1994): "Developing case-based reasoning systems: a case study in diagnosing building defects", *Proceedings of Case-Based Reasoning: Prospects for Applications*, Institution of Electrical Engineers.