Case-Agents: a Novel Architecture for Case-Based Agents

Ian Watson

AI-CBR
Dept. of Computer Science
University of Auckland
Auckland New Zealand
ian@ai-cbr.org
www.cs.auckland.ac.nz/~ian

Abstract

This paper describes a novel architecture for a case-based reasoning (CBR) system. Unlike other agent CBR systems in this architecture every case is an autonomous agent with its own retrieval, adaptation and maintenance behaviour. The paper discusses the motivation for this research and what impact this architecture has on the main tasks of processes of a CBR system. The paper concludes by describing some potential problems and benefits of the architecture.

1 Introduction

This paper describes a novel architecture for case-based intelligent agents. The research started out as a thought exercise; "*what if…*" and proceeded to a realisation that we should explore the ideas further because potentially interesting emergent properties might arise. This paper does not present the results of an implementation or of experiments, rather it is intended to stimulate other researchers to think about the architectures of their case-based reasoning (CBR) systems and perhaps to realise that things do not necessarily have to proceed as they always have done.

The paper is structured as follow: Section 2 describes some background to this work, provides the motivation for the paper and goes on to outline existing uses of agents and CBR. Section 3 outlines the case-agent architecture discussing its implications on each of the areas of representation, retrieval, reuse, revision, retention and maintenance. Sections 4 and 5 briefly discuss the relationship of this architecture to the concepts of case-coverage and of the knowledge containers. Finally section 6 concludes by describing the current status of this research and outlining potential benefits and drawbacks of the architecture.

2 CBR and Agent Technology

CBR and agent technology grew in parallel as disciplines becoming mature research areas in the 1990s. However, to an extent they progressed in isolation with relatively little work being done on case-based agents. In some ways this is surprising since both were the "hot" conference topics for a time in the 90s and one would have thought that many researchers would naturally have attempted to marry the two disciplines.

My interest was first sparked by Christopher Riesbeck's paper: *"What next? The future of case-based reasoning in post-modern AI"* [Riesbeck, 1996], which is in many ways an anti-agent technology paper arguing that "*intelligent agents are both a distant and unnecessary goal*" [Riesbeck, 1996 p.376]. Within his argument is the observation that dividing a single monolithic knowledge-based system or case-based system into communities of smaller knowledge or case-based systems adds layers of communication complexity to the entire system without necessarily improving overall functionality. This could be visualised as in Figure 1. Reisbeck's argument is controversial and one that does not seem to have had much impact given the growing popularity of agent technology.

Several researchers have gone on to discuss and implement agent systems that use CBR. Perhaps the most influential work is that of Enric Plaza and his group [Plaza et al., 1996; Plaza & Ontañón, 2001; Ontañón, & Plaza 2003] though significant contributions have been made by others (e.g., Katia Sycara's work at CMU, Robin Burke at DePaul University and Wolter Pieters at the University of Twente). This work shares common architectural themes that may be categorised as follows.

Internal-case agents
This is the most common variety, where each agent has its own internal case-base that it uses for problem solving. The agent is either given a problem to solve by a broker agent or it bids for the problem to solve. It may if necessary adapt solutions itself or pass unadapted solutions to specialised adaptation agents.

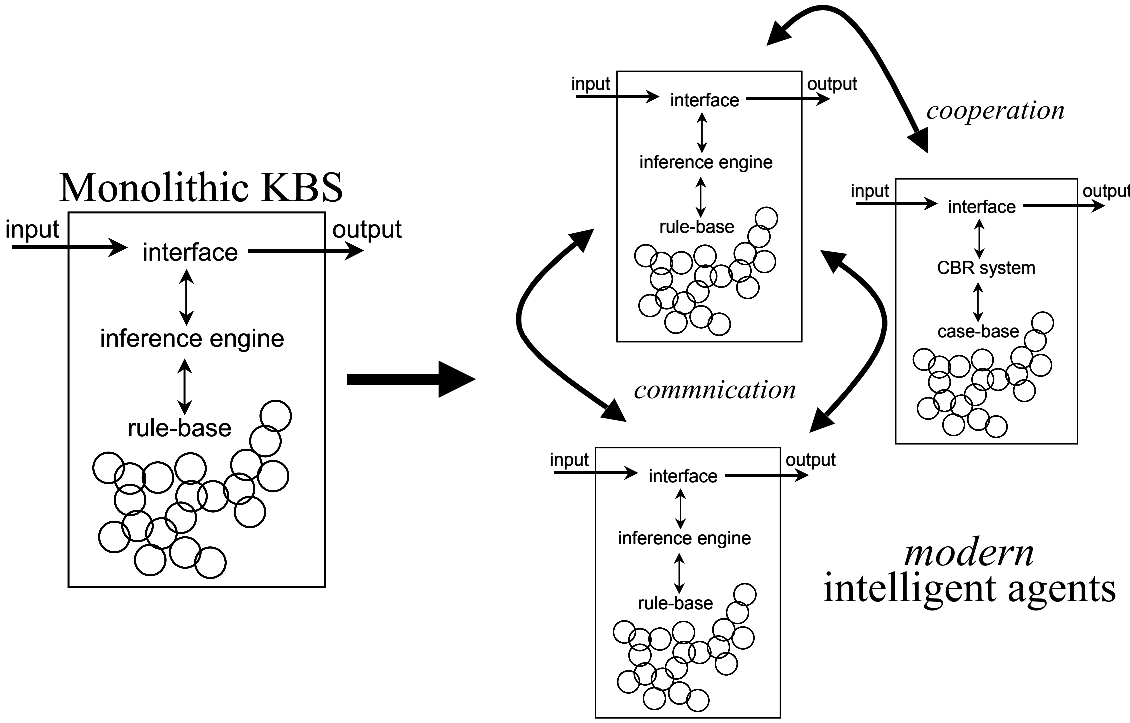

Figure 1. From monolithic KBS to communities of agents

External-case agents
Here there is one or more case-bases external to the agent that it uses for problem solving. The agent may have to decide which case-base to use to solve any given problem and again problems are assigned by a broker agent or the agent bids for it.

Intra-case agents
I have not found examples of this agent architecture in the literature, but it is entirely feasible that internal-case and external-case agents could cooperate in a single agent community. Here an internal-case agent may obtain cases from an external case base and agents may exchange and trade cases between themselves and with external case bases.

These three architectural variations seem to describe the majority of case-base agent systems research to date that I have come across. This is not surprising since they are all variations on Riesbeck's observation that intelligent agents are basically refinements of existing AI technology. The novel case-agent architecture describe below is not a refinement of existing systems and does not derive from any existing system though obviously it is influenced by current practice.

3 What if the Case was an Agent?

Think about this for a few seconds. What if the case was an agent? What if a case-base was a community of case-agents, each able to recognise if they were similar to a problem; each able to adapt their solutions, each able to maintain themselves, perhaps even to delete themselves when they became obsolete. Would this be an interesting architecture for a problem solver? We believe it would be very interesting, though we do not claim that it would be efficient, better or perhaps even as good in many situations as simpler conventional CBR architectures. The rest of this section discusses possible implications for each of the main components, tasks or processes of a CBR system.

3.1 Case representation

The basic division of a case into a tuple comprised of a problem description and a solution description would still remain. The problem description could be a simple feature:value list, a more complex structural description, a piece of text or any of the case representations CBR commonly uses. The solution could be a solution description or a method that could be used in derivational analogy. It would also be possible for cases with different internal representations to exist within the same case-base.

3.2 Case retrieval

It is in this task that the case-agent architecture becomes interesting. Each case would be responsible for assessing its own similarity to target problems. Thus, similarity would be truly local; each case-agent would be able to use it's own personal similarity metrics. Retrieval would work

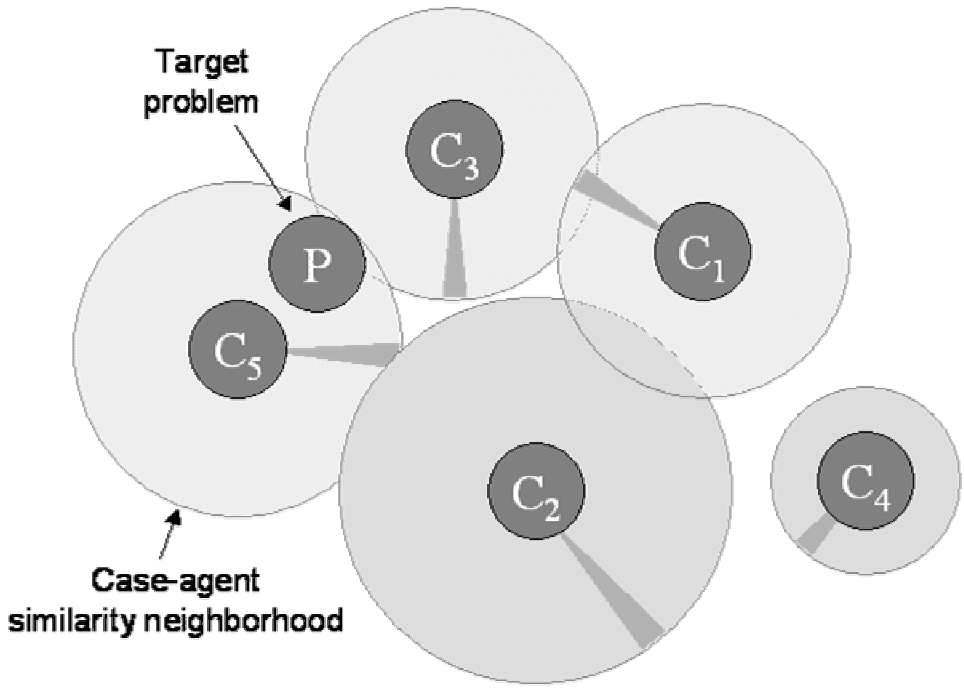

Figure 2. Five case-agents scanning the problem space

by positioning a target problem in the problem space. Each case-agent would periodically scan its neighbourhood of the problem space to ascertain if there were any target problems it might potentially be able to solve.

Case-agents could scan the problem space with differing periodicities and cases that were archived or waiting to be validated would effectively be asleep. Each case-agent would have a similarity threshold below which it would not judge itself similar to a target problem. As an analogy you might think of stationary ships using radar to detect other ships. Each ship's radar system is potentially different and may have a different range. Any vessel outside the range of the radar is invisible (i.e. not in the similarity neighbourhood). This analogy will be useful later for case maintenance as case-agents are also able to detect if other case-agents are in their similarity neighbourhood and tell how close (i.e., similar) they are.

As can be seen in Figure 2 the target problem P lies within the similarity neighbour (i.e., it is on the radar) of C_5 and C_3. It is closer to C_5 and therefore more similar.

Once a case-agent had identified that it was in the neighbourhood of a target problem several things could happen:

- The case-agent could notify a brokering agent that it was available to potentially solve the problem. It would pass the brokering agent the measure of it's similarity to the target problem and other information such as an estimate of adaptation cost/effort and its solution application history. The brokering agent may obtain offers of problem resolution from several case-agents and so would be able to calculate a global similarity taking into account if relevant factors such as adaptation cost and solution quality. Once the broker agent had decided which case-agent would be allowed to solve the problem it would instruct the selected case-agent to solve the problem or pass the case-agents solution to another specialised adaptation agent. Once can imagine situations where numerous case-agents would collaborate to solve a problem by reusing relevant parts of their solutions.
- The case-agent could go ahead and adapt its solution and present its solution to a brokering agent. The brokering agent would then compare solution qualities/cost before selecting a single solution.

3.3 Case revision

Case revision could proceed in several ways. Either the case-agent would be able to adapt the solution itself, or it would pass the solution to specialised agents capable of adapting the solution. Note that the adaptation agents need not be case-based though they are fulfilling part of the CBR-cycle. A case agent could retain metrics on adaptation effort and success which could be used in future by brokerage agents to help decide which cases to use to solve a solution.

3.4 Case retention

When a new solution has been generated a new case agent would be created. The new case agent would interact with existing case-agents in its neighbourhood of the problem space. The creation of a new case-agent would therefore instigate case maintenance which is discussed in the next section.

3.5 Case maintenance

Here again, as with retrieval, the case-agent architecture offers interesting opportunities to manage this process. As indicated above case-agents can be aware of their proximity to other case-agents. When a new case-agent is created its insertion in the problem space would automatically trigger case-maintenance. Case-maintenance would primarily be a local activity (i.e. in the locality of the newly created case.) However, case maintenance could propagate across the entire case-base much like ripples across a pond when a stone is thrown in. This is perhaps one of the more interesting emergent properties of the case-agent architecture since it provides a natural and dynamic way of controlling case maintenance. Let us explore some potential maintenance behaviours.

3.5.1 The lonely case-agent. If a case-agent was not in the proximity of any other case-agents and never saw any target problems, case-agents could be programmed to die. In this example the case-agent would represent a unique occurrence which does not reoccur and so is not worth keeping in the case-base.

3.5.2 Over crowded case-agents. It would be quite common for some case-agents to be in densely populated parts of the problem space Where case-agents were too crowded they could negotiate amongst themselves to prune their density. This might involve some case-agents deleting or archiving themselves. Thus, a compressed nearest neighbour retrieval could be implemented by case-agents within a certain proximity of others taking themselves offline.

3.5.3 Sparsely populated case-agents. In regions where case-agents were sparsely populated and therefore sometimes acting at the extreme range of their competence agents could collaborate to create new case-agents to populate the competence holes in the case-base [Smyth & McKenna, 1998].

3.5.4 Case-agent tuning. If a case-agent was repeatedly submitting itself to a broker agent but being overlooked in favour of other case-agents the case-agent could decide it needed tuning to improve its performance. This could take several forms, such as: adjusting its similarity metrics and feature weightings, or altering its adaptation methods. This implies that case-agents have a desire for themselves to be chosen and their solutions reused. Hence the case-agents can be seen as a competitive community.

4 Case Coverage

The ideas of case coverage put forward by Smyth & McKenna [1998] are complementary to the case-agent architecture. Since case-agents are aware of their relationships with neighbouring agents they would be able to form competence groups. Case-agents would also be able to recognise if they or other case-agents were *pivotal* to the competence group or merely *supporting*. Case-agents could also carry out footprint-based retrieval [Smyth & McKenna 1999] by communicating between themselves.

5 The Knowledge Containers

Richter [1995] posited that knowledge resided in four *containers* in a CBR system: the case-representation, case-vocabulary, similarity metrics, and adaptation methods. Knowledge is distributed between the containers and can be moved from one to another. In the case-agent architecture this still holds true, except that knowledge is now explicitly located with each case rather than globally. This would not seem to have any profound implications except that perhaps other agents could use this knowledge to reason about the cases.

6 Conclusions

This research started as an idea and has now progressed to several exploratory implementations. At AI-CBR we are still exploring alternative ways of implementing the case-agent architecture and must admit that none of the partial solutions implemented so far are in any way computationally efficient since much processing that would be done once globally in conventional CBR system is repeated locally for every case-agent. Since there are clearly going to be severe computational overheads to the case-agent architecture what then are the potential benefits?

Since each case-agent can have it's own (perhaps unique) similarity metrics and adaptation methods and can adjust and tune its performance the case-agent architecture is inherently flexible. A conventional CBR system with globally prescribed similarity metrics, feature weightings etc. assumes that the problem and solution spaces are uniform. As we know this is often not the case. Thus, the case-agent architecture could usefully be applied when the problem or solution space was not uniform.

Aha [1998] posited that the use of ensembles of retrieval algorithms might improve the accuracy of CBR systems. The case-agent architecture is a way of implementing ensembles since different case-agents in close proximity could have very different retrieval algorithms.

Diversity has become a popular subject in recent years recognising that for certain CBR systems (recommender systems in particular) retrieving similar cases is only part of the problem. A user often wants to see a diverse set of solutions in the retrieval set. Diversity could easily be supported in the case-agent architecture through the activities of broker agents selecting diverse solutions from the case-agents.

In many applications (again particularly in recommender systems) case solutions have a limited shelf life. For example a holiday or theatre tickets should be sold before the date the holiday or theatre show commences. A case-agent could be aware of its lifespan and adjust its behaviour accordingly. Thus, a case-agent that represented a holiday might relax its similarity threshold as the date of the holiday's commencement approached. It might also adapt the cost of the holiday downwards (i.e. discounting it) or adapt the holiday by adding other features such as free child places. In an extreme example one could imagine a case-agent whose holiday commenced tomorrow discounting its price and setting it's similarity threshold to infinity, thereby offering itself as a potential solution to every client or broker seeking a holiday.

This last example perhaps illustrates the potential benefits of the case-agent architecture, namely flexibility with encapsulation. Individual cases can alter their behaviour in anyway the developer can imagine and program without necessarily changing the behaviour of other cases. Thus CBR systems using case-agents could exhibit very dynamic and self organising properties.

Finally, I would like to conclude with a reminder that this paper is intended to stimulate discussion. The ideas described here may have no eventual utility but they will have been useful if they encourage CBR researchers to think about alternative architectures for CBR.

7 References

Riesbeck, C.K. (1996). What Next? The Future of Case-Based Reasoning in Post-Modern AI. In, Case-Based Reasoning: Experiences, Lessons & Future Directions. Leake, D.B. (Ed.). AAAI Press / MIT Press, pp.371-388.

Plaza, E. & Ontañón, S. (2001), Ensemble Case-based Reasoning: Collaboration Policies for Multiagent Cooperative CBR. In Case-Based Reasoning Research and Development: ICCBR 2001,. Lecture Notes in Artificial Intelligence 2080, p. 437-451. Springer-Verlag

Plaza, E., Arcos, J.L. & Martín, F. (1996); Cooperation Modes among Case-Based Reasoning Agents. Proceedings of the ECAI'96 Workshop on Learning in Distributed AI Systems. Budapest, pp. 90-99. (IIIA-RR-96-10).

Ontañón, S. & Plaza, E. (2003), Collaborative Case Retention Strategies for CBR Agents. In D. Bridge and K. Ashley (Eds.) Advances in Case-Based Reasoning. Proc. ICCBR 2003, Lecture Notes on Artificial Intelligence 2689, p. 392-406.

Smyth, B. & McKenna, E. (1999). Footprint Based Retrieval. In proceedings of the third International Conference on Case-based Reasoning. Munich, Germany. pp 343-357. Springer Verlag

Smyth, B & McKenna, E. (1998) A Portrait of Case Competence: Modelling the Competence of Case-Based Reasoning Systems. Proceedings of the 4th European Workshop on Case-Based Reasoning. Dublin, Ireland. pp208-220. Springer Verlag.

Richter, M. (1995). The Knowledge contained in Similarity Measures. Invited talk at ICCBR95, http://www.cbr-web.org/documents/Richtericcbr95remarks.html.

Aha, D. W. (1998). The Omnipresence of Case-Based Reasoning in Science and Application. Knowledge-Based Systems, 11(5-6), 261-273