Activating CBR Systems through Autonomous Information Gathering

Christina Carrick and Qiang Yang
Simon Fraser University
Burnaby, BC, Canada, V5A 1S6
(ccarrick)(qyang)@cs.sfu.ca

Irene Abi-Zeid and Luc Lamontagne
Defense Research Establishment Valcartier
Decision Support Technology
2459, boul. Pie XI, nord
Val Belair, Quebec, Canada, G3J 1X5
(irene.abi-zeid)(luc.lamontagne)@drev.dnd.ca

Abstract. Most traditional CBR systems are passive in nature, adopting an advisor role in which a user manually consults the system. In this paper, we propose a system architecture and algorithm for transforming a passive interactive CBR system into an active, autonomous CBR system. Our approach is based on the idea that cases in a CBR system can be used to model hypotheses in a situation assessment application, where case attributes can be considered as questions or information tasks to be performed on multiple information sources. Under this model, we can use the CBR system to continually generate tasks that are planned for and executed based on information sources such as databases, the Internet or the user herself. The advantage of the system is that the majority of trivial or repeated questions to information sources can be done autonomously through information gathering techniques, and human users are only asked a small number of necessary questions by the system. We demonstrate the application of our approach to an equipment diagnosis domain. We show that the system integrates CBR retrieval with hierarchical query planning, optimization and execution.

1 Introduction

Case-based reasoning (CBR) has enjoyed tremendous success as a technique for solving problems related to knowledge reuse. Many examples can be found in the CBR literature [17, 18, 12, 11, 21]. One of the key factors in ensuring this success is CBR's ability to allow users to easily define their experiences incrementally and to utilize their defined case knowledge when a relatively small core of cases is available in a case base.

Despite the tremendous success, traditional uses of CBR have limited its potential. In previous research, most interactive CBR retrieval systems often involve few users [1] who provide most of the answers to queries in order to retrieve

cases. In its most common mode, a CBR system involves just one user, who provides most, if not all, of the necessary information for feature values in order to perform similarity-based retrieval. For example, in a typical help desk operation, a call-center customer service representative (CSR) often enters a conversational mode, in which questions are answered by the customer, and entered by the CSR by hand. This style of interactive problem solving is important, but nevertheless is not the only mode in which to utilize a CBR system.

Our aim is to develop a more autonomous framework in which answers to CBR questions can be gathered automatically from multiple information sources. The motivation for our work derives from the evolution of an industrial-strength CBR system CASEADVISOR, developed by the CBR group at Simon Fraser University [24]. It allows a help desk organization to capture and reuse the experience and knowledge of its most experienced help desk and customer support personnel in a knowledge database that is easy to build, maintain and use. CASEADVISOR represents a typical interactive CBR (or conversational CBR [1]) application. After a user enters a natural language description of a problem, a set of cases that closely match the description is retrieved. These cases are interactively evaluated by a user based on a set of questions associated with them. When a user provides an answer to a question, a nearest neighbor algorithm is used to re-rank all retrieved cases and their associated questions in order to obtain the currently most relevant cases. The process is repeated until the user finds the target case.

Our observation is that much of the interactive question-answering process can in fact be automated. This is because many answers are available at different information sources, such as databases and web sites. In this model, a user is just one of the information sources to be queried. Following this direction, we advocate a novel approach to making such a CBR system "active" in the lifetime of an application.

In this work, the CASEADVISOR system takes up the role of a continual hypothesis evaluator. Each hypothesis is implemented as a case in the CBR system. The answers to the questions of the cases can still be obtained from the user; however, this is just one channel from which to obtain the information. We assume that there is a collection of information sources available to provide answers to the questions, or values to the attributes, in an autonomous way. We also assume that relevant information is distributed, so that no one source contains all of the information necessary to answer a question and information must be autonomously *gathered* and *integrated*. Moreover, an attribute provides only a high-level question which may need to be broken down into sub-questions and tasks by a hierarchical planning process. This task planning is adone autonomously by an information gathering sub-system. In this manner, a passive and purely interactive advisory system is turned into an active, information gathering system by using the questions in the case base as the queries to the information gathering component. We are thus inserting the information gathering component into the *Retrieve* stage of the CBR cycle, where the user may decide to *Reuse* the case at any point where the retrieval is deemed adequate.

This extension to CASEADVISOR has been implemented in JAVA to facilitate access to heterogeneous data sources, and can be applied to many situation assessment domains. Medical diagnosis and scientific theory formation are good examples of the situation assessment process: given some initial information a working hypothesis (a possible diagnosis or theory) is formed. Experimentation and testing then takes place to find further evidence to confirm or refute the working hypothesis, possibly generating alternative working hypotheses in the process. We have so far applied this situation assessment model to a military search and rescue domain and a Cable-TV equipment diagnosis domain. In the former, an initial indication about a missing aircraft will activate a case retrieval and evaluation process, in which various information sources are consulted in a continual manner [23]. In the equipment diagnosis area, again initial indications of an equipment failure will prompt the retrieval of most relevant hypothesis through a CBR retrieval process. A subsequent information gathering process will allow different hypothesis about the equipment fault to be more accurately assessed, and in the assessment process, part of the problem may be fixed. We will highlight the equipment diagnosis area later in our paper.

Our work makes contributions to case based reasoning research in several aspects. First, the model represents a method in which one can turn a passive CBR system into an active CBR system, thus increasing "interactive efficiency"[1]. A second novelty of the system is that a CBR system is used as an *information task generator* to generate information gathering tasks in an autonomous manner. Many well-known CBR systems [4,5,13] assume that the values are known for the attributes of retrieved cases. However, in situation assessment tasks, many values are not known. Therefore there is a need for verifying and retrieving these values through sophisticated query planning. Third, since we assume an open system architecture in which many information sources are expected to coexist, the system integrates a CBR component with an information gathering component. The information-gathering component performs global task expansion, planning and optimization.

Well-known CBR systems can be enhanced by our model. Systems such as HYPO [4,5] and CASEY [13], rely on problem descriptions that are collections of attributes and values to retrieve similar cases. Cases in their case bases are also assumed to have their attribute-values ready for comparison with the incoming problem description. However, these systems do not emphasize on *how* these attributes and values are obtained. Our approach nicely complements these and other CBR systems in that an autonomous model is provided for gathering information in order to execute case based reasoning. In addition, our system complements case based planning (CBP) systems [20] in that, while CBP systems adapt a plan case after a similar plan is identified, in our approach "information plans" are adapted *during the retrieval process* in order to find the most similar case.

The organization of the paper is as follows. We first present a system overview in Section 2. Then we discuss in Section 2.1 the case base representation and the case retrieval process. In Section 2.2 we describe how to use the system to

generate questions and how we select one of those questions for execution. We then discuss in Section 2.3 how to use a task planning and execution module to gather the information. Finally, in Section 3, we present a practical example of how our system can reduce the number of questions posed to the customer in a cable TV call centre domain. We conclude the paper with discussions of related works, our future research plans and conclusions.
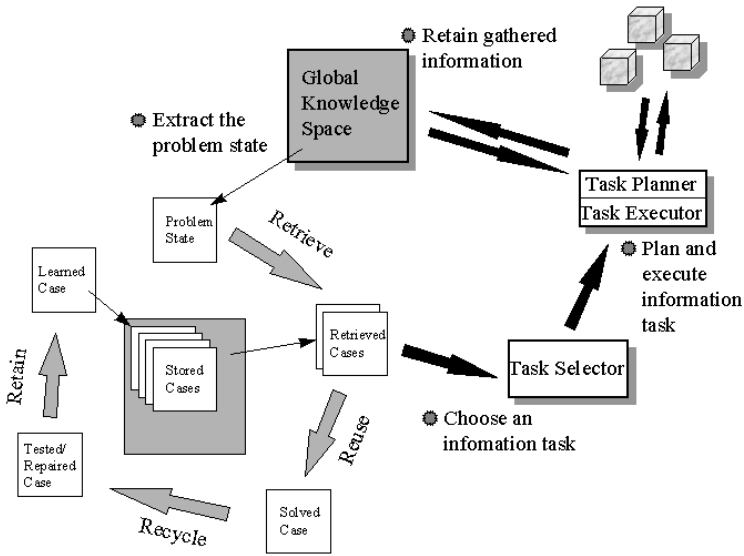
2 System Overview

Fig. 1. The situation assessment cycle intersects the CBR cycle in the Retrieve stage.

As shown in Figure 1, our situation assessment system is made up of several complex modules which interact internally in a cyclical fashion as well as externally with a number of data sources, and intersecting the CBR cycle in the Retrieve phase. At the initiation of the cycle, the Global Knowledge Space (GKS) contains all relevant information which is known by the user and has been entered into the GKS. From the GKS is created a problem state, which is used to query the CASEADVISOR case base. This constitutes the situation assessment portion of the system, in which two or more competing hypotheses are produced which best explain the information contained in the GKS. These retrieved hypotheses then form the context for the information gathering task. The Task Selector uses the working hypothesis (that which best explains the data) as well as the competing hypotheses to formulate a set of all questions which may further

distinguish the hypotheses. The Task Selector then chooses from that set one question which is to be executed as an information task, and the Task Executor plans and executes that task using the available data sources. (Decomposition of an information task results in **sub-tasks** and possibly **tasks** which may need to be performed in the process of answering the information task.) Should the question be unanswerable, control returns to the Task Selector module which chooses a new question as an information task for the Task Executor. When the Task Executor has obtained an answer to the question, the information is placed in the GKS, an updated problem state is created and used to query the case base, and the set of competing hypotheses is thus re-evaluated.

2.1   Situation Assessment

The situation assessment in our system is provided by a case-based retrieval system. The case base stores previous situations and their associated attributes as cases. A case in our system is defined as a tuple $< H, S, T >$, where H is a textual description of the diagnosis or *hypothesis* of the situation, S is a set of one or more <question,answer,weight> tuples representing a typical *state* leading to H, and T is a conjunction of zero or more *tasks* which may be executed should H be the working hypothesis. These tasks in T are different than those information tasks generated by the unanswered questions from S in that they are not intended to aid in the accuracy of the situation assessment and are merely things which should be done if H is the working hypothesis (such as "Notify supervisor.").

Hypothesis: parental control switch on

Question	Answer	Weight
problem description	poor reception of the cable signal	1.00
channels affected	specific channels	0.70
uses parental control	yes	0.80
has cable box	yes	0.40
outlets concerned	1	0.30

Fig. 2. A sample case from the cable TV call centre domain.

In our cable TV call centre domain, a case consists of a problem cause (the situation) and its associated conditions and effects (the state) as shown in Figure 2. The cable TV case base consists of a number of these cases, and a problem cause can be identified by the given exemplar state. The information contained in the GKS is used to formulate a problem state - the current state represented as a set of existing conditions and/or effects. This is currently accomplished by extracting from the GKS all statements relating to any of the questions in the case base. This problem state is used by the retrieval system to find the cases in the case base which most closely match the problem state using a k-nearest-

neighbour algorithm. The problem cause in each case retrieved then becomes a possible hypothesis.

2.2   The Task Selector

Once we have retrieved the most likely hypotheses for a problem state, one question must be chosen to become an information task for execution. This selection involves estimating the costs of performing the information tasks, combining this estimation with the estimated information quality of the questions, and optimizing (possibly trading off) the two.

Information Quality When deciding which high-level information task should be executed to refine the situation assessment, it is useful to take into account how much information can be gained by its execution. It is desirable to maximize information gain so that a working hypothesis can be confirmed or refuted as quickly as possible, with as few questions as possible. As noted in [1], using the information gain formula typically found in decision tree induction (as in [19]) is not feasible when the case space is sparse. For this reason we have used an estimation of information gain which we will denote *information quality* for clarity.

To begin, we consider the information quality for only those questions which are unanswered in the set of retrieved cases. This allows us to eliminate questions which may be unanswered but are irrelevant to the current context (where the context is defined by the set of competing hypotheses). We currently measure information quality by considering three factors which would seem to have some influence on the importance of a question:

- the number of times the question appears in the retrieved cases,
- the weights of the question in the different cases, and
- the ranks of the cases containing the question.

The last two heuristics have previously been used in our CASEADVISOR system. In the current study all of these factors were given equal weighting in calculating the information quality, though we are also studying the effects each of these factors typically has on the information quality of a question. Questions are then ranked according to their estimated information quality, and this ranking can be further utilized in the task selection process.

Estimating Cost of Query Execution The cost of executing an information task far outweighs the cost of planning such a task [10]. It is therefore a worthwhile endeavor to take the time to find a good plan for gathering information. Different data sources may have different monetary costs for accessing them, are able to respond in differing amounts of time, etc. For example, the plan fragment shown in Figure 3 gives four possible locations for finding the answer to the high-level task: the customer profile, the customer accounts database, the
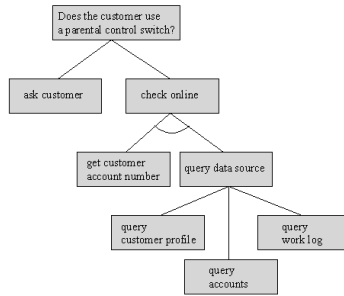
Fig. 3. A partial expansion tree for the query "Does the customer use a parental control switch?"

log of installation and maintenance work, and the customer herself. Checking the work log may incur a high time cost if the data is remotely stored, while local customer profiles may be old or incomplete. Each source has various cost constraints, some of which may be more important than others. The problem is then to find the best execution plan, which minimizes the cost of answering that question.

In order to estimate the cost of answering a question, it is necessary to formulate an execution plan. We accomplish this task by creating a hierarchical task network (HTN) [22,7] from a library of task schemata[1]. This HTN presents an explicit parent-child relationship on information tasks, where a sub-task can either be directly executed, or can be decomposed into its constituent parts. It also contains any temporal constraints, user-imposed preferences and cost estimates which are important in the estimation of the cost of answering a question. It is also possible to be able to perform a task (or some part of it) in a number of different ways. An execution plan for an information task can then be seen as a constrained AND/OR graph of decompositions (conjunctions) and alternatives (disjunctions). The HTN query plan represented by the AND/OR graph can be different for different situations and information state in the data sources.

Figure 3 shows a partial AND/OR tree for the execution of an information task to find out whether a customer uses a parental control switch. Any path through this tree which covers all children of each encountered AND node and at least one child of each encountered OR node is a possible plan to execute the information task. These schemata contain not only information pertaining to the AND/OR structure of a plan, but

Our system currently estimates execution cost using a mini-max network flow algorithm on its query expansion network. All leaf nodes in AND/OR trees (directly executable queries) have associated with them a cost value which is a function of time cost, dollar cost and reliability cost. For each candidate query, the Task Selector examines all leaf nodes in its AND/OR tree. The cost values

---

[1] For brevity we do not show the schemata here.

are propagated upward to the root node, taking the sum at AND nodes and the minimum at OR nodes. It then remains to traverse the tree from the root node, taking the minimum at each OR node. The resultant traverse of the tree is a minimum cost solution and the leaf nodes in this traverse form a low-cost query execution plan. The topic of planning and tree traversal will be visited again in Section 2.3.

Combining Information Gain and Execution Cost In choosing an information task (or question) for execution, it is important to pay attention to both the information quality of the question and also the estimated cost of that question. If a question has a very high information content but it is unlikely that we will obtain an answer before final decisions must be made, then perhaps a faster question with slightly less information quality is in order. Thus there can be a trade-off between information quality and execution cost. Currently, our system uses hard-coded parameters in a function of gain and cost, though we are considering the benefits of having user- or domain-specified priorities on the different constraints involved.

2.3   The Task Planner and Executor

The Task Planner and Executor module performs the usual information gathering tasks of planning, optimizing and executing an information task. It receives the information task which has been chosen by the Task Selector module and searches that task's AND/OR tree for the least-cost plan. Since each leaf node specifies how it is to be executed, the leaf nodes in this solution plan are executed by invoking the functions, modules or agents whose calls are contained therein. This AND/OR tree search algorithm returns a low-cost solution plan for answering the question when the AND/OR graph is a tree (i.e. an executable task does not show up twice in the graph). When information sources are shared by different tasks, the optimization problem of finding a lowest-cost solution plan is NP-hard; we are currently experimenting with different heuristic algorithms for solving it.

   If all possible solutions to the task fail (i.e. all children of the top-most OR node have been exhausted and no solution has been found), then control returns to the Task Selector module to choose an alternate question for answering. This process continues until a question has successfully been answered or until there are no more question options. In the latter case, processing halts and the working hypothesis is presented as an assessment of the situation. When a question has been successfully executed, the answer to the question is placed in the GKS. This triggers the creation of a new problem state to be formed from the contents of the updated GKS. The problem state is presented to the case-based retrieval system, and hypotheses are retrieved with new matching scores based on the updated input state. This cycle continues until there are no more questions which can aid in the accuracy of the case retrieval, or until halted by the user.

3   Sample Scenario

To demonstrate the utility of our system we show here a sample problem from the cable TV troubleshooting domain. Figure 4 shows three cases retrieved by the customer's problem description of "poor reception", and the six questions which are relevant to those retrieved cases in the order produced by our information quality heuristics.

Case I: signal problems

problem description	poor reception of the cable signal	1.00
channels affected	specific channels	0.80
local signal	clear	0.95

Case II: customer not subscribed to package

problem description	poor reception of the cable signal	1.00
channels affected	$29 - 58$	0.75
channels affected	$29 - 43$	0.75
channels affected	$44 - 58$	0.75
subscribed	no	0.50

Case III: parental control switch on

problem description	poor reception of the cable signal	1.00
channels affected	specific channels	0.70
uses parental control	yes	0.80
has cable box	yes	0.40
outlets concerned	1	0.30

Q1   Which channels are having the problem?
Q2   Is the picture clear on the local set?
Q3   Does the customer use a parental control switch
Q4   Is the customer subscribed to these channels?
Q5   Does the customer have a cable box?
Q6   How many outlets have the problem?

Fig. 4. Cases relevant to the problem description "poor reception", and the questions relevant to those cases.

Figure 5 shows the results of a diagnostic session in which a customer has poor reception on certain cable channels. There were five data sources available in our cable TV domain: a customer profile database, a customer accounts database, a work/installation log, a database of current signal problems, and the customer herself. Figure 5(a) gives the results obtained from the initial problem description, and shows that question Q1 has been chosen for execution. This question is posed to the customer, as we currently have no on-line means of obtaining the information. The response to this question is then added to the problem state and the new retrieval from the case base gives us the results in Figure 5(b). At this point Q2 is chosen for execution. A partial expansion for

this information task is shown in Figure 6, and the execution plan chosen is the highlighted path. Note that at node C the first path chosen was to a database of current signal problems, which resulted in failure due to the absence of the requested information at that site. It was then necessary to re-plan and chose an alternate, next-best solution. Once all of the information was obtained and integrated to answer the question, this answer was added to the problem state and again the case base performed a re-assessment.

This process continued through all of the cycles depicted in Figure 5[2]. The four on-line data sources provided answers to all questions but Q1 in a realistic fashion. This shows a tremendous potential for reducing the duration of the question-answer session with the customer. By using information gathering techniques, we also reduce the burden of information search on the call centre employee and speed up retrieval of information, thus diagnosing problems more quickly and servicing more customers in a shorter amount of time.

The sample session just presented shows how our system can be useful with a passive CBR diagnosis mechanism: when a customer recognizes that she has a problem, she can invoke the system with an input problem state and the information gathering component obtains evidence to aid in the diagnosis. But these same mechanisms can also be used in an active manner. Consider the use of active databases and monitoring agents in the information gathering component: instead of waiting for a new question to be posed, these monitors and triggers become activated whenever a relevant change occurs in a data source. We can then use this information to *foresee* a problem. The case base can continually re-rank retrieved cases, based on these changes in the data sources. We then have a system which already has much of the information needed for a diagnosis when a customer calls in with their problem description. This integrated technique can be seen as performing active case-based reasoning in a backward fashion, where the reasoning (AND/OR tree expansion) occurs from the objectives back to the information sources. In [15], we study how to combine active databases and CBR in a *forward* manner. Indeed, an active system such as this could even predict that customers will phone the call centre with a particular problem given changes in the local signal, listed outages, last-minute changes in the TV schedule, etc.

4   Related Work

In the case-based reasoning community, Conversational CBR (CCBR) has attracted substantial research [1]. CCBR, essentially interactive CBR, involves the refinement of diagnoses through interaction or conversation with the user, asking questions which are considered to have high information gain. These questions are based upon the unanswered attributes in the problem case which are relevant to the retrieved cases, and are ranked according to some heuristic such as the number of cases in which the attribute occurs. Popular in help-desk applications, commercial tools such as Inference Corporation's CBR Express exemplify

---

[2] We had turned off the solution threshold, allowing re-assessment to take place until there were no further questions to be answered.

initial input:	poor cable reception	
retrieved:	signal problems	45
	customer not subscribed to package	27
	parental control switch on	27
chosen question:	Which channels are having the problem?	
sources queried:	customer	

(a)

retrieved:	signal problems	64
	parental control switch on	36
chosen question:	Is the picture clear on the local set?	
sources queried:	GKS, signal problems database, local monitor	

(b)

retrieved:	parental control switch on	56
	signal problems	44
chosen question:	Does the customer use a parental control switch?	
sources queried:	GKS, customer profile database	

(c)

retrieved:	parental control switch on	65
	signal problems	35
chosen question:	Does the customer have a cable box?	
sources queried:	GKS, customer profile database	

(d)

retrieved:	parental control switch on	69
	signal problems	31
chosen question:	How many outlets are having the problem?	
sources queried:	GKS, customer accounts database	

(e)

retrieved:	parental control switch on	71
	signal problems	29

(f)

Fig. 5. Results of a diagnostic session in which a customer has poor reception on certain cable channels. Selected questions are planned and executed among the various data sources, and the increasing amount of known information leads to an increased accuracy in the diagnosis.

CCBR [1]. These systems attempt to find the quickest way to increase the accuracy of diagnosis through estimating information gain. Further research has used model-based inferencing to reduce the number of questions asked of the user by eliminating redundant questions [3]. These systems still remain user-guided however, and therefore need not consider certain problems introduced by IG such as cost estimation and planning information tasks. [2] discusses ongoing research in which state information can be collected from users and also from sensors, leading into research in optimizing cost and gain estimations.

CBR and HTN planning have also been integrated in the NaCoDAE/HTN system [16]. There, however, the CBR is used to interactively generate plans, where these plans are constrained by the HTN. This puts the NaCoDAE/HTN system at the *Revise* stage of the CBR cycle, which is complementary to our
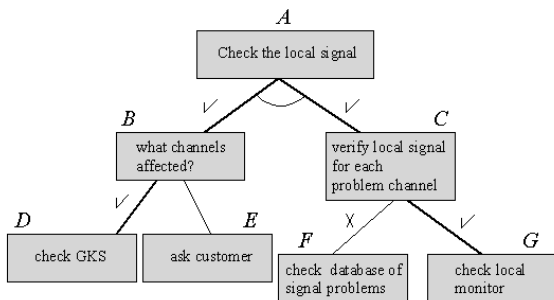
Fig. 6. The task expansion for the information task "Check the local signal."

work. A similar system to NaCoDAE/HTN is our CASEADVISOR system as described in [24], which attaches a case with a decision tree. When a candidate case is identified, the decision tree is evaluated, prompting the user with a series of questions and actions to follow. The decision tree search essentially performs the case adaptation work.

In the area of automated information gathering, many researchers have been investigating methods of reducing the cost of executing a query through query reformulation and optimization [14, 9, 8, 6, 10]. This is a query in the database sense, and corresponds to a single question which must be answered through the information gathering component in our work. These generally involve assigning various costs (dollar cost, time cost, accuracy cost, etc.) to data sources and reasoning to minimize those costs. For example, the SIMS system models subsumptive relations which are useful in query formulation, and also facilitates descriptive models of resources for query optimization. The InfoSleuth system is the result of intensive research in ontologies, and also uses a frame-based, three level representation to provide a detailed model of the domain and resources [9]. In more flexible systems such as BIG [14], users are able to specify which costs are more important to them and the cost-minimizing function takes this into account. Thus, the BIG system optimizes IG plans according to user preferences. Given the quality, cost and time features of accessing the various available data sources, BIG attempts to optimize these features with respect to a set of user-specified constraints on the features. This allows for different definitions of "optimal", depending on the user of the system.

5   Future Work

With the support of the preliminary results obtained, we are eager to investigate aspects of our system further. Several variants of the task selection and query planning algorithms are worth investigation, taking into account different constraints and optimizing those constraints. Global optimization of the planning process should take into account various cost, content overlap and information

gain information of the information sources. However, this discussion is beyond the scope of this paper. We have several proposals for improving the cost estimation/optimization algorithm and are performing such studies:

- rearranging sub-queries within planning constraints to minimize data source accesses,
- studying the effects of global optimization as opposed to local optimization, and
- learning and/or estimating the cost of a new data source.

Also under investigation are the impacts information quality and execution-time have on the time to converge to an assessment. Perhaps a question with high information content would be worthwhile executing even if the execution time was estimated to be very high, if it would allow convergence to a single situation assessment hypothesis soon afterwards. Discovering a relationship between information quality, execution time and time to convergence would allow us to create a cut-off where we could say that the information quality is not worth the execution time.

Uncertainty in gathered information is another aspect which we consider important in future versions of our assessment system. At present, all information inserted into the GKS is given a weight of 1.0, though there is the option available to weight the information from the GKS with degrees of certainty. Using this option will impact not only the maintenance of the GKS, but also our task selection algorithm as it may be desirable to verify already "known" information which has a low certainty.

We also see as important the problem of displaying to the user all of the useful information about the assessment process, without being obtrusive. Since our program is meant to act as a support to the user, working quietly in the background, we do not wish to overload the user with process information. Indeed, it is information overload which our system attempts to overcome. However, the user should feel in control and this involves giving the user access to the various processes which are taking place within the system. Even if it is not an option for the user to override a system decision, it can make new users more comfortable with the system to simply know the data upon which plans are being made.

6   Conclusions

Using automated information gathering to aid in situation assessment is a novel research topic. We have combined a case-based retrieval system with information gathering techniques which results in a fewer number of questions posed to the user. The advantage of the system is that the majority of trivial or repeated queries to information sources can be done autonomously through an agent-like system, and human users are only asked a small number of necessary questions by the system. Under this model, we can also use the CBR system to continually generate questions that are planned for and executed based on information sources such as databases and the Internet, resulting in an active diagnostic

system. In the cable TV call centre domain, this concept shows a tremendous potential for reducing the duration of the question-answer session with the customer. By using information gathering techniques, we also reduce the burden of information search on the call centre employee and speed up retrieval of information, thus diagnosing problems more quickly and servicing more customers in a shorter amount of time. Our system also goes a long way to facilitating self-diagnosis via the internet or an automated phone system, further reducing call centre costs.

Acknowledgments

References

1. D. Aha and L. Breslow. Refining conversational case libraries. In *Proceedings of the Second International Conference on Case-based Reasoning (ICCBR-97)*, Providence, RI, July 1997.

2. D. W. Aha, L. A. Breslow, and T. Maney. Supporting conversational case-based reasoning in an integrated reasoning framework. Technical Report AIC-98-006, Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, DC, 1998.

3. D. W. Aha, T. Maney, and L. A. Breslow. Supporting dialogue inferencing in conversational case-based reasoning. Technical Report AIC-98-008, Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, DC, 1998.

4. K. Ashley. *Modelling legal argument: Reasoning with cases and hypotheticals*. MIT Press, Bradford Books, Cambridge, MA, 1990.

5. K. Ashley and E. Rissland. A case-based approach to modeling legal expertise. *IEEE Expert*, 3(3):70–77, 1988.

6. O. M. Duschka and A. Y. Levy. Recursive plans for information gathering. In *Proceedings of IJCAI-97*, Nagoya, Japan, August 1997.

7. K. Erol, J. Hendler, and D. S. Nau. Htn planning: Complexity and expressivity. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 1123–1128, Seattle, WA, 1994. AAAI Press/The MIT Press.

8. M. R. Genesereth, A. M. Keller, and O. M. Duschka. Infomaster: An information integration system. In *Proceedings of the ACM SIGMOD Conference*, May 1997.

9. R. J. B. Jr., W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, and D. Woelk. InfoSleuth: Agent-based semantic integration of information in open and dynamic environments. In *Proceedings of SIGMOD'97*, 1997.

10. C. A. Knoblock, Y. Arens, and C.-N. Hsu. Cooperating agents for information retrieval. In *Proceedings of the 2nd International Conference on Cooperative Information Systems*, Toronto, Canada, 1994. University of Toronto Press.

11. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publisher, Inc., 1993.

12. J. Kolodner and D. Leake. a tutorial introduction ot case-based reasoning. In D. Leake, editor, *Case-Based Reasoning:Experiences, lessons & Future Directions*. American Association for Artificial Intelligence, 1996.

13. P. Koton. Reasoning about evidence in causal explanation. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, Cambridge, MA, 1988. AAAI Press/MIT Press.

14. V. Lesser, B. Horling, F. Klassner, A. Raja, and T. Wagner. Information gathering as a resource bounded interpretation task. Technical Report 97-34, University of Massachusetts Computer Science, March 1997.

15. S. Li and Q. Yang. ActiveCBR: Integrating case-based reasoning and active databases. Technical Report TR 1999-03, School of Computing Science, Simon Fraser University, Burnaby BC Canada, January 1999. http://www.cs.sfu.ca/ qyang/Papers/activecbr.ps.gz.

16. H. Muñoz-Avila, D. C. McFarlane, D. W. Aha, L. Breslow, J. A. Ballas, and D. Nau. Using guidelines to constrain interactive case-based htn planning. Technical Report AIC-99-004, Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, DC, 1999.

17. T. Nguyen, M. Czerwinski, and D. Lee. Compaq quicksource—providing the consumer with the power of ai. *AI Magazine*, 1993.

18. A. Perini and F. Ricci. An interactive planning architecture: The forest fire fighting case. In M. Ghallab, editor, *Proceedings of the 3rd European Workshop on Planning*, pages 292–302, Assissi, Italy, September 1995. ISO Publishers.

19. J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

20. M. Veloso, H. Munoz-Avila, and R. Bergmann. General-purpose case-based planning: Methods and systems. *AI Communications*, 9(3):128–137, 1996.

21. I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers, Inc., 1997.

22. Q. Yang. Formalizing planning knowledge for hierarchical planning. *Computational Intelligence*, 6, 1990.

23. Q. Yang, I. Abi-Zeid, and L. Lamontagne. An agent system for intelligent situation assessment. In F. Giunchiglia, editor, *Proceedings of the 1998 International Conference on AI Methodologies, Systems and Applications (AIMSA98)*, volume 1480 of *Lecture Notes in AI*, pages 466–474, Sozopol, Bulgaria, September 1998. Springer Verlag.

24. Q. Yang, E. Kim, and K. Racine. Caseadvisor: Supporting interactive problem solving and case base maintenance for help desk applications. In *Proceedings of the IJCAI 97 Workshop on Practical Applications of CBR*, Nagoya, Japan, August 1997. International Joint Conference on Artificial Intelligence, IJCAI.