Case-Based BDI Agents: an Effective Approach for Intelligent Search on the World Wide Web

Cindy Olivia, Chee-Fon Chang, Carlos F. Enguix, and Aditya K. Ghose

Decision Systems Laboratory Department of Business Systems The University of Wollongong
Northfields Avenue, Wollongong NSW 2522, Australia
co01@uow.edu.au, c03@uow.edu.au, cfe01@uow.edu.au, aditya@uow.edu.au

Abstract

This paper describes a framework that integrates case-based reasoning capabilities in a BDI agent architecture as well as its application to the design of Web information retrieval agents. The research proposed in this paper generates two key insights. First, it shows that the integration of case-based reasoning in a BDI agent architecture is a non-trivial exercise that suggests interesting ways of building BDI agents with learning capabilities. Second, it demonstrates the efficacy of the resulting framework by presenting the design of intelligent Web information retrieval agents that are effective in well-demarcated domains.

1. Motivation

One of the main objectives of this research is to prove that the integration of case-based reasoning (CBR) with BDI (Belief-Desire-Intention) agent models improves the performance of currently deployed Web Information. Some of the full text-based Web IR systems support a set of advanced search features under the typical section known as "power search". These features may support domain constraining (i.e. uow.edu.au), page depth-level control, word stemming, and search of related terms with the aid of universal dictionaries and thesauri. However, as we have observed, currently deployed Web IR systems have a series of limitations, which we cite as follows:

- **Inefficient crawling:** the lack of ability to reuse knowledge from previous traversals that implies redundant exhaustive search in many cases.
- **Limited domain-specific knowledge:** none of them take the advantage of domain-specific knowledge associated to well-demarcated logical domains on the WWW to support automated search guidance towards promising sites.
- **Limited concept spanning:** few of them are capable of retrieving concepts that span more than a single Web page. For instance, the research interests associated to an academic member's homepage may be specified in another Web page. Therefore, unless we take into account the relationship between these Web pages, we will not be able to find and extract information relating to both the research interests associated to an academic member and the academic member's data itself.

The approach proposed in this research intends to solve the limitations described above by supporting intelligent real-time search based on the ability of the agent to learn from previous cases stored in a case memory.

The rest of the paper is organised as follows: section 2 presents the preliminaries of BDI agent architecture, case-based reasoning, and related work. The formal framework of CBR-BDI architecture is presented in section 3. Section 4 describes in detail the architecture and actual implementation of Web CBR-BDI agents, a prototype as a proof-of-concept in a well-demarcated domain such as university Web pages. Finally, we conclude this paper by revisiting the essential features of Web CBR-BDI agents.

2. Preliminaries

2.1 BDI Agents

BDI agents have been widely used in relatively complex and dynamically changing environments. BDI agents are based on the following core data structures: beliefs, desires, intentions, and plans (Rao and Georgeff 1995). These data structures represent respectively, information gathered from the environment, a set of tasks or goals contextual to the environment, a set of sub-goals that the agent is currently committed, and specification of how sub-goals may be achieved via primitive actions. The BDI architecture comes with the specification of how these four entities interact, and provides a powerful basis for modelling, specifying, implementing, and verifying agent-based systems.

2.2 Case-based Reasoning

Case-based reasoning (CBR) has emerged in the recent past as a popular approach to learning from experience, and has been implemented in a variety of systems including IR. Case-based reasoning (CBR) is a type of reasoning based on the reuse of past experiences called cases (Kolodner 1993). Cases are description of situations in which agents with goals interact with the world around them. Cases in CBR are represented by a triple (p,s,o), where p is a problem, **s** is the solution of

the problem, and o is the outcome (the resulting state of the world when the solution is carried out). The basic philosophy of CBR is that the solution of successful cases should be reused as a basis for future problems that present a certain similarity (Kolodner 1993). Cases with unsuccessful outcomes or negative cases may provide additional knowledge to the system, by preventing the agent from repeating similar actions that leads to unsuccessful results or states.

2.3 Related Work

It has been pointed out by (Marko and Shoham 1995) that learning is needed to improve system's performance. In the area of Web IR, system's performance can be measured in terms of efficiency (speed of finding solution) and the quality of solution. Previous studies concerning the integration of CBR in IR commonly leverage the intelligence of knowledge-based CBR with the applicability of IR systems, resulting in the improvement of the search performance. One of such application is the Analog Devices' a CBR-based electronic catalog accessible through the Web (Volrath, Wilke and Bergmann 1998). The deployment of CBR with similarity measures and domain-specific knowledge-base has an advantage over other retrieval techniques that use strict retrieval rule where inexact matches are rejected. In other words, if the user requests for an information can not be fully satisfied, the system still provide something close to the request instead of nothing. This interesting feature is not only applicable for the improvement of online-catalog in the Web, but also for many other Web IR systems in general. Web IR agents, with intelligence and learning capabilities to effectively find alternative solutions related to the user queries will be a great contribution to overcome many of the limitations of currently deployed Web-IR systems.

3. The Formal Framework of CBR-BDI Agents

3.1 Data Structures of CBR-BDI Agents

Situated in a world or an environment, an agent senses and effects its environment. The agent is denoted by a 5-tuple < B,D,I,PL,KB >. The formal definition of the agent's structure is presented below:

Definition 1. Belief B corresponds to the description of the agent's environment <E>. To accommodate learning from previous experience, we modify the informational state of the agent by adding a case memory <CM>. We denote an agent's belief as B =< E,CM>.
Environment E described by a set of values for a fixed set of non-null environment variables, denoted as E= $<e_1,e_2,...e_n>$ where $e_i \in E$ and $i \in \{1,2,...n\}$, hence for any instance of time, E will describe the current environment.

- Case memory CM stores sets of previous cases, formulated as a 4-tuple < B_m, D_m, I_m, O_m > representing past beliefs, past desires, past intentions, and the outcome value respectively. An occurrence of intention formation corresponding to a desire in a belief-world, is stored as a case in the case memory. The distinction between successful and unsuccessful cases are made through the assignment of outcome value, where cases with an outcome value lower than a certain threshold or value are considered as unsuccessful. This mechanism enable agent's to learn from previously successful cases, and at the same time avoid making previous mistake.

Definition 2. Desire D corresponds to tasks allocated to the agent, also regarded as a set of current goals (G) in a particular belief-world. D is denoted as < $G_1,G_2,...G_n$ > where $G_i \in G$ and $i \in \{1,2,...n\}$.

Definition 3. Intention I represents a trigger for a corresponding plan from the plan library PL. Intentions are organised in a list based on similarity proximity, where the further the Is apart, the less similar they are.

Definition 4. Plan library PL is a collection of plans (P). Each plan is a pair denoted by <p,A> where p is the plan descriptor that describes the pre and post conditions for the execution of plan. A is the plan body which is compiled of a set of actions to achieve an intention denoted by < $a_1,a_2,...a_n$ > where $a_i \in A$ and $i \in \{1,2,...n\}$.

Definition 5. Knowledge Base KB is composed of a set of default rules associated with case retrieval, case adaptation, and case update process. These set of default rules also include a set of rules that handle exceptions; an occurrence where no similar cases stored in the case memory.

3.2 Case-based Reasoning Processes of CBR-BDI Agents

In general, a CBR problem-solving cycle consists of case retrieval, adaptation, and case updating. Case retrieval is a process of finding and retrieving a case or a set of cases in the case memory that is considered to be similar to the current problem. Case adaptation is a process where the solutions of previous similar cases with successful outcomes are modified to suit the current case, bearing in mind the lessons learned from previous similar cases with unsuccessful solution. Case update process is the process of updating case in the case memory or insertion of new case in the case memory. Once a solution is decided and acted upon, the outcome of the action will be stored along with a description of how successful the outcome was. The following sub-sections will describe the case-based reasoning processes of CBR-BDI agent.

3.2.1 Case Retrieval Process. The *similarity function* S takes as input:

1. The current set of Bs
2. The current set of Ds
3. The case memory CM.

and produces as output :
a set of similar cases $\{C_1, C_2, \ldots C_n\}$ where each $C_i \in CM$.

$$S_b : B \times CM \rightarrow C$$

$$S_d : D \times CM \rightarrow C$$

$$C' = (S_b(B, CM) \cap S_d(D, CM))$$

$$S(C) = F(C')$$

S_b : function to retrieve cases with similar B
S_d : function to retrieve cases with similar D
B : the set of all possible sets of beliefs
D : the set of all possible sets of desires
C : the set of similar cases
C' : the set of cases with similar Bs and Ds
S : the similarity function
CM : the set of all possible sets of cases
F : filtering function that filters out the elements of C' which have:

1. Bs that render current Ds unachievable
2. Ds that are unachievable given current Bs

For exceptional situation, where no cases are retrieved during the case retrieval process, the adaptation process will be skipped, and the standard default rules are fired to generate an intention.

3.2.2 Case Adaptation Process. The *adaptation function* A takes as input:

1. A set of cases C retrieved during the case retrieval process

$$A : C \rightarrow I$$

and produces as output : an Intention I

The adaptation function is used to adapt cases retrieved during the case retrieval process. Cases in the list(C) may be categorised as good or bad cases. Adaptation process involves choosing the best case from the list (C). The intention of the chosen case will be adapted based on similarity proximity. The adapted intention is compared with the intention of the bad cases. If the adapted intention is similar to the bad cases' intention, the adapted intention will be dropped and the next best case will be adapted. This process will be repeated until an adapted intention is found to be different from the bad cases' intention. If the adaptation is unsuccessful, where none of the adapted intention is different to the bad cases' intention, then a default rule is fired.

3.2.3 Case Update Process. Once an intention is executed, an *outcome function* is performed. The executed intention I and outcome value O along with the belief B and desire D will be stored as a new case in the case memory CM.

The *outcome function* O_f is used to generate a quantitative evaluation value of the success or failure of the particular intention.

$$O_f : I \rightarrow O_m$$

3.3 Algorithm of CBR-BDI Agent Interpreter

CBR-BDI Interpreter

```
Initialise-State();
REPEAT UNTIL END (B)
  Update B(Event-Queue);
  Case-Based-Reasoning(B, D);
  Execute I();
  Store-Executed-Attitudes(Case-Memory);
  Drop-Executed-Attitudes();
  Drop-Impossible-Attitudes();
  Get-New-External-Events();
END-REPEAT

/* Case Retrieval Based on Similarity Function */
Case-Based-Reasoning(B, D);
  Option-List := S(B,D)
  IF Empty(Option-List) THEN
        Selected-Option := (Default r);
  ELSE
        Option-List := Sort(Option-List);
        A(Option-List);
  END-IF

/* Similarity Function */
S(B,D)
  REPEAT UNTIL END(CM)
        B-List := Sb(Bm);
  END-REPEAT

  REPEAT UNTIL END(CM)
        D-List := Sd(Dm);
  END-REPEAT

  REPEAT UNTIL END(B-List, D-List)
        Option-List := Duplicate(B-List, D-List);
  END-REPEAT

  Option-List:= Eliminate(Unachievable(Option-List));

/* Adaptation Function */
A(Option-list)
  REPEAT UNTIL END(Good(Option-List))
        Selected-Option:= Adapt(Select(Option-List));
  END-REPEAT
```

```
IF Unsuccessful(Adapt) THEN
        Selected-Option := (Default r);
END-IF
Update-Intentions := Selected-Option;

/* Update Case Memory */
Store-Executed-Attitudes
  Update-Case-Memory(B,D,I,O);
```

4. Architecture of Web CBR-BDI Agents

This section presents the CBR-BDI agent architecture for information retrieval on the WWW. One of the main objectives of this research is to prove that the resulting integration improves the performance of currently deployed Web IR systems in terms of search efficiency and resource discovery in well-demarcated domains. Our approach demonstrates the intelligent search capabilities of such agent based on the ability to learn from previous cases stored in a case memory, coupled with a static domain-specific knowledge base.

The case memory is tightly coupled with a static domain-specific knowledge-base contextual to the logical domain / application domain of interest. In other words, the agent can be adapted to any kind of logical application domain provided the relevant domain-specific knowledge-base is included. This integration not only accelerates the IR process but also enhances resource discovery, finding a wider range of related pages that might not be retrieved using current Web IR systems. As a proof-of-concept by demonstration, Web CBR-BDI agents are designed to locate and extract information from homepages of academic staff members with particular research interests.

4.1 Core Components of Web CBR-BDI Agents

We can distinguish as core components in Web CBR-BDI Architecture: the Case Memory, the Domain-Specific Knowledge-Base and the CBR-BDI Interpreter (presented in section 3.3). The subsequent section presents in greater level of detail the core components of CBR-BDI agent architecture (see Figure).

4.1.1 Domain-Specific Knowledge-Base. The domain-specific knowledge-base is implemented in a form of concept hierarchy. The concept hierarchy is represented as collection of keywords representing broad areas of expertise (concepts). Attached to these concepts, a collection of keywords (sub-concepts) representing specific sub-area of expertise. Concepts are mapped to specific university academic-entities on a well-demarcated application domain such as Australian universities (Enguix et al. 1998). In the current implementation, we focus the concept-hierarchy only in the area of information technology, entirely based on the IEEE Internet Computing Classification Index (IEEE 1998) and the ACM Computing Classification System (ACM 1998). The concept hierarchy plays an important role in focusing the search process in the start-up of the Web CBR-BDI agent where no previous cases are stored, and when the similarity function does not allocate any particular case in the case memory. Furthermore, it helps the system to identify related research interest if it fails to retrieve an exact match information. For example, if no exact match for research interest in mobile agents is found, the system is able to retrieve academic homepages with relevant research interest such as intelligent agents, autonomous agents, etc.

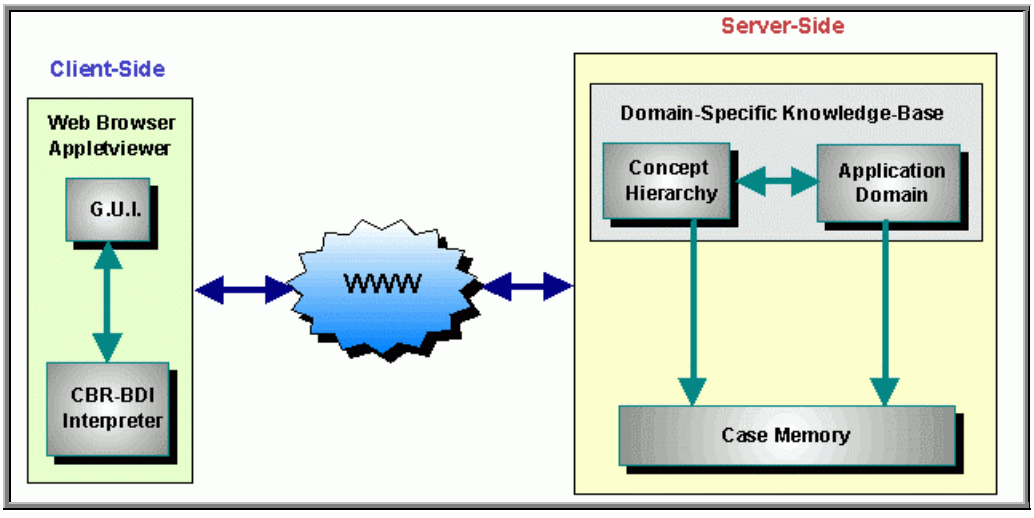

Figure 1. Core Components of Web CBR-BDI Agents

Belief	University Domain to be searched
Desire	Sub-Concept/Concept represents the specific/similar research interest being searched
Intention	Focused search to concept-related Academic Entities: faculties, departments, etc.
Outcome Status	Status successful/unsuccessful case: true/false (0/1)
Outcome URL	URL Staff Link Directory Academic Entity

Table 1. Core Data Structures of the Case-Memory

4.1.2 Case Memory. Cases stored in the case memory are constructed in terms of data structures presented above (see table 1).

Negative cases indicate that the previous intention was unsuccessful in trying to find the specific research interest in a target academic entity at the university domain. Given that similar cases are sorted by outcome status (found/not found), the Web CBR-BDI agent first scans the most promising URLs (outcome status = found), and leaves for the last stages of the search the less promising ones (negative cases).

We are particularly interested in extracting the information at the level of academic homepages. Academic homepages have a higher probability of modification in terms of content. As a matter of fact, we only store the URL of staff-link directories within the scope of target academic-entities captured by previous traversals. This strategy permits us to deliver "fresh information" from real time traversing. Moreover, it minimises space storage requirements in the case memory, as we do not store the URL of each particular homepage where the specific research interest was found.

In the case where the similarity function retrieves similar cases, the case memory may lead directly to a promising URL from where to initiate either a depth-first or breadth first search, instead of traversing exhaustively the sub-webs of a particular university. The similarity function scans case memory data structures in the following prescribed order (see figure 2):

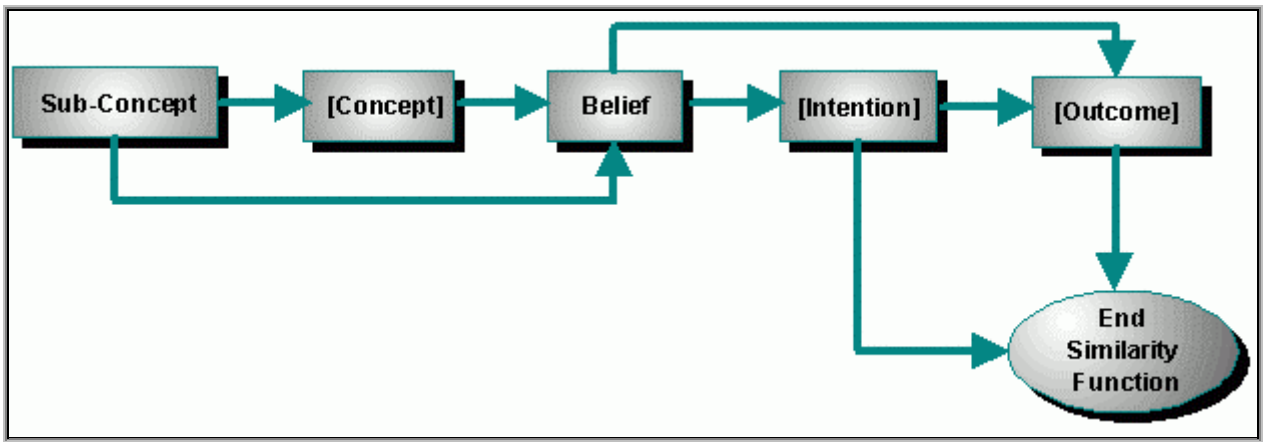

Figure 2. Case Retrieval Process Based on the Similarity Function**. Note:** [] indicates optional

The following table (table2) depicts the possible combinations between data structures involved in the retrieval of cases based on the similarity function to derive current intention to be executed.

COMPARISON OF CURRENT CASE WITH CASE MEMORY							CURRENT CASE
Case	Desire Sub-concept stem	Desire Concept-name	Belief University	Intention Academic-Entity	Outcome-Status-Found +/-	Outcome-URL-Staff-Directory	Current Intention
1	✓	✓	✓	☹	TRAVERSAL ORDERED BY:+ Cases - Cases	GET	☞ URL-Staff Link Directory
2	✗	✓	✓	☹	NO ORDER	GET	☞ URL-Staff Link Directory
3	✗	✗	✓	✓	☹	GET	☞ URL-Staff Link Directory
4	✗	✗	✓	✗	☹	☹	☞ URL of Related Faculty Homepages
5	✓	✓	✗	GET INTENTION FROM CASE 1	☹	☹	☞ Root URL to Staff Link Directories Related Academic Entities
6	✗	✓	✗	GET INTENTION FROM CASE 2	☹	☹	☞ Root URL to Staff Link Directories Related Academic Entities
7	✗	✗	✗	GET INTENTION FROM CONCEPT-HIERARCHY	☹	☹	☞ Root URL to Staff Link Directories Related Academic Entities

LEGENDA:	
✓	SIMILAR
✗	DISTINCT
☹	IGNORED
☞	START TRAVERSING FROM
GET	RETRIEVE STORED VALUE

Table 2. Case Comparison Process to Define Current Intention

4.2 Web CBR-BDI Agent Implementation

As the development of the Web CBR-BDI agent is still in progress, we have not come to the state to conduct the full evaluation function in terms of performance comparison with other Web IR systems. However, the current prototype developed in JAVA™ run with several tests data has proved the features claimed in the proposed architecture.

The filtering and extraction of information performed by the Web CBR-BDI agent is based on pattern string matching heuristics, and a rule-based extraction system, assigning word-stems for each sub-concept. By assigning word-stems for each particular sub-concept, we extend the range of related pages retrieved in addition to those retrieved based on an exact match criteria.

4.3 Overall Process

The main objective of Web CBR-BDI agent is to retrieve information related to academic members with specific research interests in selected university domains. End-users are presented with a simple and intuitive GUI (see figure 3) where they can specify sub-concepts/research interests associated to a given concept/domain of knowledge, together with the

universities of interest. Web CBR-BDI agents are triggered by pressing the start button. The first objective of the agent is to perform a standard CBR analysis of the input problem case. The input problem case is constructed by the combination of an end-user's selected university domain (belief), and sub-concept/specific research interest (desire). The similarity measures serve to find the most similar cases with the input problem case. The results obtained from the CBR analysis drive the Web traversal of the agent to retrieve the desired information.

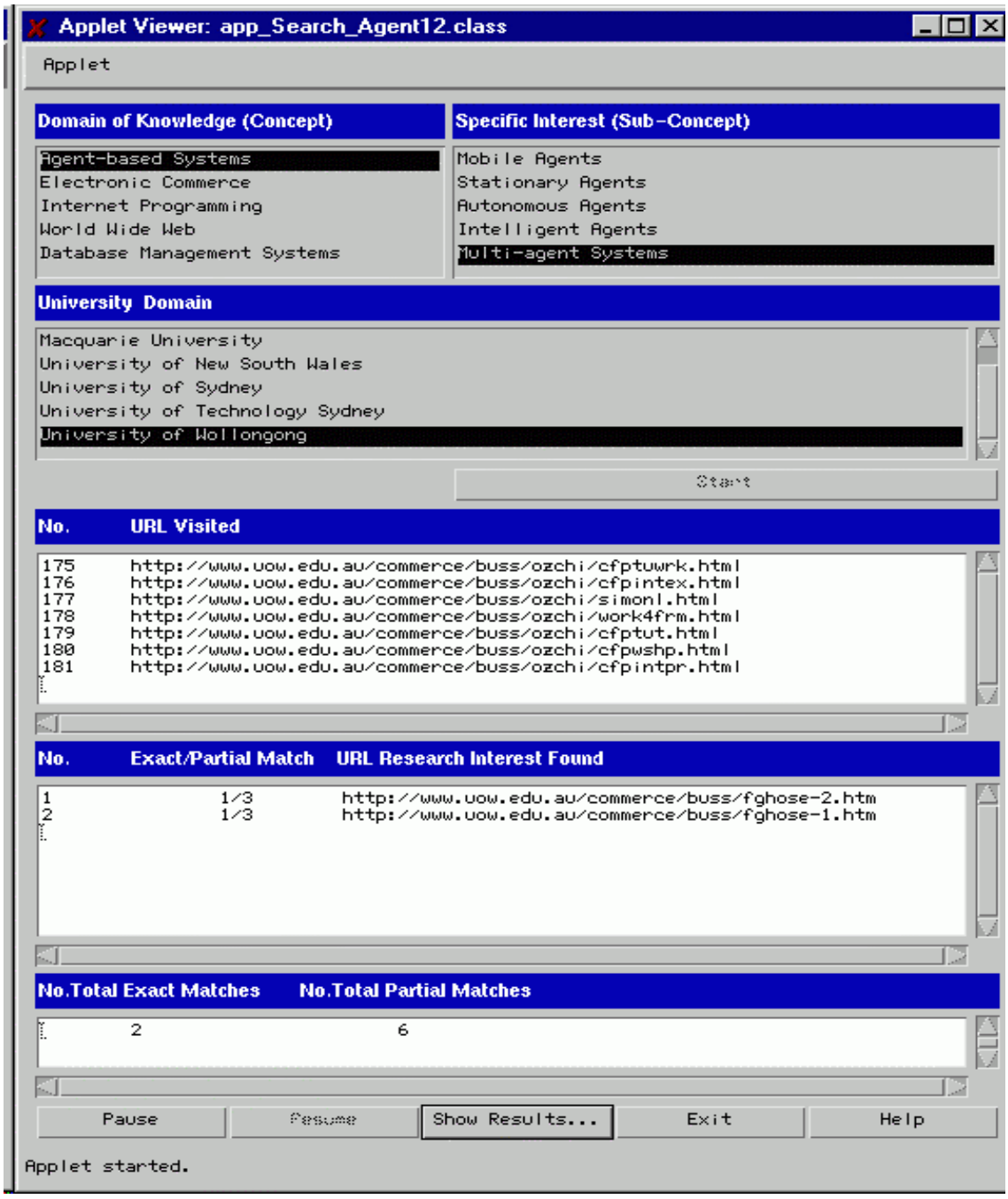

Figure 3. GUI of the Web CBR-BDI Agent

5. Conclusion

Current implementation of Web CBR-BDI has proved the efficacy of the proposed approach in overcoming the limitations of current Web IR systems by exhibiting the following features:

- Capability of learning and reusing previous knowledge from cases stored in the case memory. This learning capability is essential in optimising the search process, avoiding redundant or exhaustive traversal of sub-Webs and focusing the search to the most promising Web sites.
- Capability of getting search direction, even in the worst scenarios, when no case exist in the case memory or when the similarity function does not retrieve any similar case in the case-memory. This intelligent feature is supported by the domain-specific knowledge-base.
- Capability of shifting automatically its operation mode from breadth-first search to depth-first search at certain stages conditioned by the current intention being handled by the CBR-BDI interpreter. This capability is an essential feature to perform a focused and staged search process (i.e. first find faculties and associated staff-link directories in breadth-first mode and from each staff-link directory initiate a depth-first search) permitting the location and extraction of concepts (research interest) that may span more than a single Web page.

Another interesting feature of this architecture is the flexibility supported by the domain-specific knowledge-base component. Potentially, this component can be adapted to any particular well-demarcated application domain on the WWW. Such adaptation would require minimum modifications with respect to the rest of components in the architecture.

6. References

ACM, "The ACM Computing Classification System", May 1998
Available at: http://www.acm.org/class/1998/ccs98.txt (ASCII file)

Enguix, C.F., Davis, J.G., Ghose, A.K. 1998. Database Querying on the World Wide Web. Technical Report TR98/5/101.Decision Systems Laboratory, University of Wollongong
Available at:
http://budhi.uow.edu/postgrad/carlos/tr98_5_101.htm

IEEE Internet Computing online, 1998. IEEE Internet Computing online Classification Index. October 1998.
Available at: http://www.computer.org/internet/xtras/

Kolodner,J.L., 1993. *Case-Based Reasoning*. Morgan Kauffman, Publishers Inc.

Marko, B., and Shoham,Y. 1995. Learning Information Retrieval Agents: Experiments with Automated Web Browsing. In AAAI-95 Spring Symposium on Information Gathering.

Rao, A.S. and Georgeff, M.P. 1995. BDI Agents: From Theory to Practice. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), San Fransisco, USA.

Volrath, I., Wilke, W., Bergmann, B. 1998. Case-based Reasoning Support for Online Catalog Sales. *IEEE Internet Computing*, July-August: 47-54.