The Exchange of Retrieval Knowledge about Services between Agents

Mirjam Minor, Mike Wernicke

Institute of Informatics
Humboldt University Berlin
Unter den Linden 6
10099 Berlin
minor@informatik.hu-berlin.de
wernicke@informatik.hu-berlin.de

Abstract: Recently, several work has been done in hybrid case-based reasoning (CBR) systems [Ah98] [Le99]. Hybrid CBR systems are case-based systems that are combined or integrated with either one or more other AI methodologies (for example neural networks or rule-based systems) [Le99]. Some systems combine agents with case-based reasoning: Models of the users in form of profiles are managed as cases [MLR02] or as sets of cases [CCH02]. [WL98] describe how CBR can be used for situation assessment in a multi-agent soccer system; [RA98] performs e-mail filtering with an agent using CBR as a service. In this article, we present a hybrid system of assistant agents using textual case-based reasoning (TCBR) for a smart retrieval of services. The service library of an agent is regarded a case base. As agents are allowed to exchange services, the exchange of parts of their retrieval knowledge - namely information entities and local similarity relationships - has to be discussed and implemented. The integration of the received tiny case retrieval nets with the own retrieval structures is a crucial task of our work described here. Slightly different duplicates of information entities and similarity relationships have to be handled as well as inconsistently assigned strings. In the future, semantic aspects of the harmonization of merged background knowledge could be taken into account as it has been done in the ontology community.

1. Introduction

Recently, several work has been done in hybrid case-based reasoning (CBR) systems [Ah98] [Le99]. Hybrid CBR systems are case-based systems that are combined or integrated with either one or more other AI methodologies (for example neural networks or rule-based systems) [Le99]. Some systems combine agents with case-based reasoning: Models of the users in form of profiles are managed as cases in [MLR02] or as sets of cases in [CCH02]. [WL98] describe how CBR can be used for situation assessment in a multi-agent soccer system; [RA98] performs e-mail filtering with an agent using CBR as a service. Experience management (EM) systems can increase their capabilities enormously when they exchange their knowledge. In this paper, we present an hybrid approach of EM agents that use CBR to improve the interpretation of the users' orders. The agents collaborate by the exchange of parts of their individually collected treasury of experiences. This includes the knowledge contents that are managed for the users as well as the knowledge how to retrieve and apply this knowledge. In Section 2, we describe the scenario of personal assistant agents that manage experiential knowledge how to perform daily tasks with the computer and are able to apply and execute those experiences autonomously for their users. Section 3 contains the case-based approach to extend the original scenario by a better textual user interface. Section 4 deals with the exchange of individually collected knowledge, gives the technical background for this, and provides solutions of some problems concerning the merging of personal Case Retrieval Nets. Section 5 handles some open research issues. Finally, in Section 6 we discuss some related work and give a summary of this paper.

2. The Scenario of Personal Assistant Agents

In 1999, Kühnel et al. developed a system of personal assistant agents that distribute how-to-knowledge in form of services [Ku99] [Ku00]. Services are utility programs like 'print a file' or 'perform a query on the Web'. A personal assistant agent manages a set of such services for it's user. By order of the user the assistant agent is able to execute a service. If necessary, a planning component helps the agent to combine subservices to a bigger service. The agent handles its user's requests either autonomously or – if necessary - in collaboration with agents of other users. Agents can ask other agents to perform a certain service or even learn services from others. For instance, if an agent has got the order to print a dvi file but knows only how to print a postscript file, it can look for another agent to transform the dvi format into postscript. In case one of the other agents knows the UNIX command dvi2ps, the original agent can either ask to learn this service or send the dvi file to the owner of the service and let it do the tranformation remotely. User-defined access and executable rights control the exchange of services and the execution of services for other people.

The services are implemented in Java. The executable code is embedded in a service description (see Fig. 1) that specifies the category of the service, a free text that explains the user what the service is aimed to do, the needed access rights, and some further properties of the service. In a second level, the service developer can define execution conditions as well as input and output parameters. It is crucial for the usability of the assistant that it quickly finds a service description that matches the user's request. The original system uses a simple string matching algorithm to interpret the commands that are given by the user. This requires that the user knows exactly the vocabulary of the service descriptions. That condition is often not fullfilled and therefore the user has to click through the tree of service descriptions to find a service. In the approach presented in this paper, the matching of user input and service base is performed case-based (see Section 3).

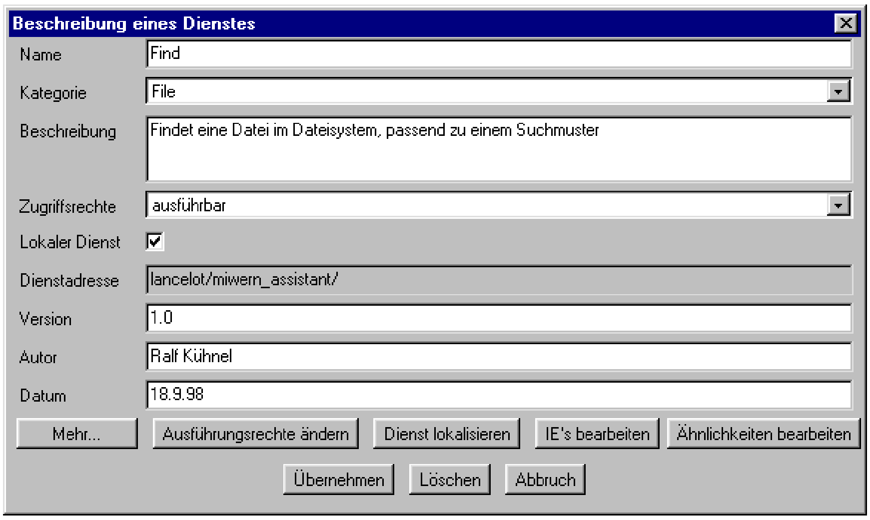

Fig. 1. Example of a service description for the service 'find'

3. Service Descriptions as Textual Cases in a Case Retrieval Net

A case-based approach supports the interaction of assistant agents and users. The textual part of the service descriptions is regarded a case in the case base each. The user input is asked as a query to the case base. The agent responds to the user with the retrieved services. They are ordered in a list of best matching services to the user's input. So the user can select an appropriate service from the list and give the agent the order to execute it.

The retrieval uses a bag of words approach for textual case-based reasoning (TCBR). We will just sketch this approach in the following; for detailed information about TCBR we refer to [LHK98]. Important terms of the texts are mapped on sets of so-called information entities (IEs). The mapping includes stemming and handles abbreviations and different spellings. For instance, the service description opens Yahoo in a Netscape Navigator is mapped on the set of IEs *Case1* = {*__open__*, *__yahoo__*, *__Netscape Navigator__*} while __open__ collects strings like open, opens, opening and so on. The query is treated in the same way and compared with the particular cases by means of a compositoric similarity function *SIM*. A local similarity function *sim* specifies similarity values between the information entities like *sim*(*__open__*, *__start__*) = 0.3 or *sim*(*__Netscape Navigator__*, *__Internet Explorer__*) = 0.5. An example of a very quick and simple function *SIM* – which we could extend slightly (see Section 5) - is a sum of the concerned local similarity values:

$$SIM(Query, Case) = \sum_{e_i \in Query} \sum_{e_j \in Case} sim(e_i, e_j). \qquad \textbf{(1)}$$

The retrieval is implemented in a Case Retrieval Net (CRN) [LB96]. When the assistant agent is started, the case retrieval net is built from the individual service library and two dictionaries with information entities and local similarity relationships. The user can upgrade the agent's abilities by new services, new vocabulary and new similarity relationships.

The advantage of this case-based retrieval is not only that it solves the paraphraze problem, i.e. finds also a service if the user chooses other vocabulary to describe her wishes than the author of the service. Additionally, it provides services that are similar to the query what gives a bigger choice of services. For example, if the user asks to open Yahoo with a Netscape Navigator, the agent can also offer to open Google with an Internet Explorer. This might be useful when only the alternative Web browser has been installed on the current operating system. In the prototype of the agent, the user has to check manually whether the execution conditions of a service are satisfied. In case they are not, the assistant agent presents an error message.

4. Personal Case Retrieval Nets

The agents' service libraries are stored distributedly as each agent has its individual set of services. So, it is obvious to organize the retrieval processes and the according dictionaries also autonomously for each agent. A central case retrieval net in the manner of yellow pages would have been easier to manage. But to send each query to a central agent to let it search appropriate services implements rather a client-server model than an agent system. Furthermore, as the agents can specialize in particular domains to follow their users' interests and abilities, it could lead to inconsistencies and ambiguities in a net that tries to represent all of them. So, we prefer to use personal case retrieval nets on the basis of initially delivered and individually extendable dictionaries.

When an agent sends a service and the according description, the service description might contain important terms that are not yet known to the receiver agent. We decided to let the agent transfer some parts of its background knowledge to enable the receiver agent to retrieve this service for the user in future. The sender agent appends all information entities on the message that belong to the service description including the similarity relationships defined within this subset of IEs. The receiver agent filters out the new ones and integrates them with its personal dictionaries as well as the new service description with the service library. The merging of the background knowledge - the original dictionaries and the tiny other dictionaries - might lead to some types of conflicts:

A) non-identical duplicates of information entities
B) duplicates of similarity relationships with different weights
C) differently assigned strings

The conflicts A) and B) are resolved quite simply: Double IEs are merged to one IE whith all strings belonging to both originals. Double similarity relationships are resolved by choosing the maximum degree of the original similarities. This is a bit arbitrarily, it would have been also possible to use the minimum value or to keep the own similarity value, but being specified in both originals is a kind of emphasis and we reward it by the higher value. To handle conflicts of type C) is a bit more complicated. For instance, if we have originally an IE *__print__* with the strings *print* and *printer*, but a new IE *__printer__* with the strings *printer* and *print device*. To decide semantically which IE should keep the string *printer* is easy but to let it decide a program is difficult. As a straight-forward solution, we allow both IEs to keep their strings with the result that both can be activated when this string appears in a text. Even worse, there might be defined a similarity relationship in the new CRN between *__print__* and *__printer__*. More sophisticated solutions could ask the user for conflict resolution. In our approach, the user does not have any maintenance effort when her agent learns a new service from another agent. Of course, when defining a new service the dictionary of information entities has to be maintained. And to avoid conflicts of type C),we recommend to model the IEs as simply as possible, i.e. to map only strings with the same stem to one IE and to separate nouns and verbs in different IEs. Of course, this leads to a higher effort in defining similarity relationships.

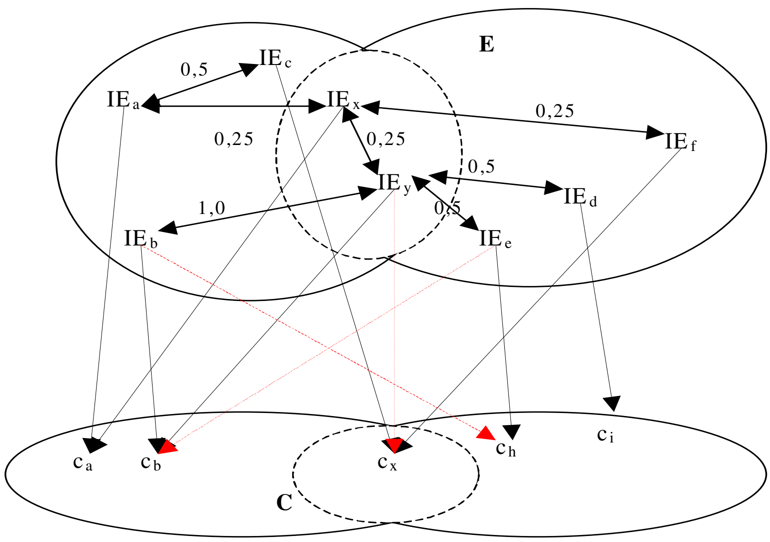

Fig. 2. Merged Case Retrieval Nets

The Case Retrieval Net has to be rebuilt to be able to retrieve the new service in future user requests. The new net can be regarded a merge of originally two Case Retrieval Nets, namely the agent's own CRN and a tiny received CRN consisting of just one new case node and the according sets of information entities and local similarity relationships. The above discussed merging method is also applicable on the join of bigger Case Retrieval Nets (see Fig. 2). The dotted arcs are automatically added when rebuilding the merged net.

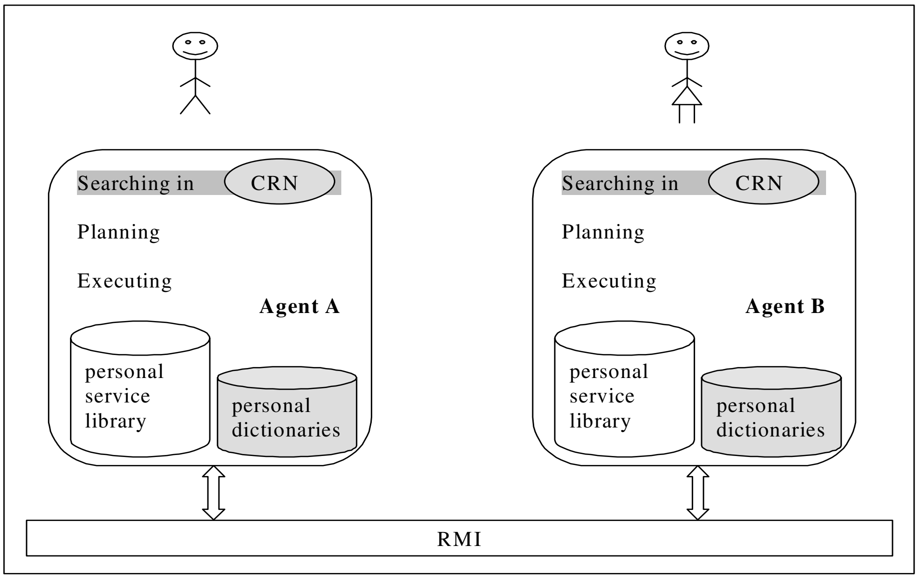

Fig. 3. The prototype's overall system architecture

5. State of Implementation and Open Issues

We have extended Kühnel's Java implementation of an assistant agent (see [Ku99], [Ku00]) with a prototype for the case-based interpretation of the user inputs (see Fig.3). The grey areas show the changed parts of the architecture. The three main tasks of the agent are still to search appropriate services, to plan bigger services as sequences of subservices, and to execute services. The original string matching component to search matching services to user requests has been replaced by a case-based retrieval method. The new Java package 'TCBR' implements a Case Retrieval Net with the dictionaries and algorithms to perform a case-based retrieval on texts. A new package 'AgentTCBRShell' integrates 'TCBR' with the source code of the agent. 'AgentTCBRShell' extends the original description of a service with a set of information entities 'individualTCs' and a set of according similarity relationships 'individualSIMs'. A service designer has to specify those sets when defining a new service. Of course, the system gives authoring support with making suggestions from its dictionaries. The prototype runs as a Java application. It uses Kühnel's communication mechanism based on RMI to exchange Java objects, namely the extended service descriptions. Receiver agents integrate the service description with their own knowledge base immediately. This includes the part of the sender's Case Retrieval Net that belongs to the textual description of the received service. The receiver extends it's dictionaries with the new knowledge.

When the agent rebuilds its Case Retrieval Net from the extended dictionaries it might be less homogeneous than carefully maintained background knowledge. Especially the problem of missing links between the new IEs and old ones in the merged CRN can spoil the future retrieval results. As an example, the agent may know the IEs __*google*__ and __*search engine*__ with similarity links in both direction. If it has received the new IE __*altavista*__ with a similarity relationship to __*google*__, there is missing a link to and from __*search engine*__. We have not yet a good solution for this problem as we don't want to bother the user with the integration of the new knowledge. To detect missing links automatically requires sophisticated algorithms. At the moment, we think about building a kind of transitive wrapper to get bridges between the new IEs and the IEs associated to their neighbors. This would either have impacts on the similarity function or require a preprocessing when building the CRN. The neighbors of the new IEs are connectors between the old and the tiny new parts of the Case Retrieval Net and should be treated in a special way.

Further open research issues are the multiple assigning of strings in the dictionaries as described in Section 4 and an evaluation of the case-based approach compared to the straight-forward string matching.

6. Discussion and Summary

Our agents cooperate not only in sharing services but also in sharing the knowledge how to retrieve and apply the services. They exchange parts of their background knowledge, i.e. service descriptions and little subnets of their individual Case Retrieval Nets. In opposite to the approach described in [Le96], the agents do not perform a collaborative retrieval actually with strictly distributed retrieval knowledge but combine individual retrieval results of possibly overlapping case bases. The integration of received retrieval knowledge with the own knowledge repository plays a major role in the presented work.

In the ontologies community, the integration and interoperability of ontologies is a recent research topic [EU02]. In opposite to our purely syntactic merging of terms and similarity relationships, there the semantic harmonization of different ontologies plays a major role. In our recent approach, terminological heterogeneity is not yet regarded as we use less sophisticated taxonomic constructs than ontologies. We express relationships between terms only by one type of relationship, namely similarity. So, the frequency of semantic problems might not be as high as in ontological systems, but it would be worth while to have a closer look on the solutions that have been worked out for ontologies. How we resolve different degrees of similarity, has been described in Section 4.

Web Services are a new tool to exchange and combine programs with XML as interface [Sc02]. As Web Services can return objects, we could employ it instead of RMI to implement the communication between agents in future. The service descriptions can be transformed to WSDL, the Web Services Description Language. RMI is sufficient in the current prototype, but in future, Web Services should be taken into consideration, as they seem to become a well-known standard with Microsoft's .NET activities. However, the handling of the service descriptions, e.g. the retrieval or execution of services, has still to be implemented within the agent. The Web Services provide just a further language to wrap functionality for communication purposes, they do not implement essentially new concepts.

In this paper, we presented an assistant agent system for how-to knowledge using case-based reasoning for the user interaction. The agents provide a case-based retrieval on their individual service libraries. This improves the retrievability of services that match a user's request enormously. Therefore, the agents build individual Case Retrieval Nets which can be merged with background knowledge received from other agents. We discussed problems of merging CRNs and provided some solutions. In our vision of the future development of agents, such autonomous learning processes by including external knowledge sources and the exchange of kow how will play a major role.

References

[Ah98] D. Aha and J. J Daniels (Eds.): 1998 AAAI Workshop on Case-Based Reasoning Integrations, AAAI Press Technical Report WS-98-15, Madison, USA, 1998.

[CCH02] L. Coyle, P. Cunningham, C. Hayes: A Case-Based Personal Travel Assistant for Elaborating User Requirements and Assessing Offers. In: S. Craw, A. Preece (Eds.): Proc. ECCBR 2002, LNAI 2416, p. 505-518, Springer Verlag, Berlin, 2002.

[EU02] J. Euzenat, A. Gomez-Perez, N. Guarino and H.Stuckenschmidt (Eds.): Ontologies and Semantic Interoperability, Proceedings of the ECAI'02 Workshop, Université de Lyon, 2002.

[Ku99] R. Kühnel: Assistant Agents that Distribute How-To-Knowledge, Proceedings of AOIS'99, 1999.

[Ku00] R. Kühnel: Eine Methode zur Entwicklung agenten-basierter Software, PhD thesis, Humboldt University Berlin, 2000.

[Le99] B. Lees (Ed.): Hybrid Case-Based Reasoning Systems, Proceedings of the ICCBR'99 Workshop, University of Kaiserslautern, 1999.

[Le96] M. Lenz: Preparing Case Retrieval Nets for Distributed Processing. In: L. Czaja, P. Starke, H.-D. Burkhard, M. Lenz (Eds.): Proc. of the Workshop on Concurrency, Specification and Programming (CS&P-96), Humboldt University, Berlin, 1996.

[LB96] M. Lenz, H.-D. Burkhard: Case Retrieval Nets: Basic Ideas and Extensions. In: G. Görz, S. Hölldobler (Eds.): KI-96: Advances in Artificial Intelligence, Springer Verlag, LNAI 1137, 1996.

[LHK98] M. Lenz, A. Hübner, M. Kunze: Textual CBR. In: Case-Based Reasoning Technology - From Foundations to Applications, Lenz M., Burkhard HD., Bartsch-Spörl B., Wess S. (Eds.), LNAI 1400, Springer Verlag, Berlin, 1998.

[MLR02] M. Montaner, B. López, J. L. de la Rosa: Improving Case Representation and Case Base Maintenance in Recommender Agents. In: S. Craw, A. Preece (Eds.): Proc. ECCBR 2002, LNAI 2416, p. 234-248, Springer Verlag, Berlin, 2002.

[RA98] Michael A. Redmond, B. Adelson: AlterEgo E-Mail Filtering Agent - Using CBR as a Service. In: D. Aha and J. J Daniels (Eds.): 1998 AAAI Workshop on Case-Based Reasoning Integrations, Madison, USA, 1998.

[Sc02] H. Schulz: Weltweites Werkeln. c't 06/02, p. 236 ff., Heise Verlag, 2002.

[WL98] J. Wendler, M. Lenz: CBR for Dynamic Situation Assessment in an Agent-Oriented Setting. In: D. Aha and J. J Daniels (Eds.): 1998 AAAI Workshop on Case-Based Reasoning Integrations, Madison, USA, 1998.