Dynamic retrieval of heterogeneous knowledge sources in a Multi-Agent-System

Pascal Reuss[1], Kerstin Bach[1,2] and Klaus-Dieter Althoff[1,2]

[1]Intelligent Information Systems Lab
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany

[2]Competence Center Case-Based Reasoning
German Research Center for Artificial Intelligence (DFKI) GmbH
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
kerstin.bach@dfki.de, reusspa@uni-hildesheim.de, klaus-dieter.althoff@dfki.de

Abstract: This paper presents a *Multi-Agent-System* that uses Case-Based Reasoning (*CBR*) to dynamically assess the required knowledge sources for querying heterogeneous, distributed knowledge sources. We describe the idea of the so called *Knowledge Line* and its implementation in form of a *Multi-Agent-System* . Our prototype is realized in the domain of travel medicine where preventive information about destinations, diseases and potential health risks are provided. Further we present the results of an evaluation carried out with the travel medicine prototype and discuss related work on the aspects of *Multi-Agent-Systems*, *CBR* and scalability of such systems.

Keywords: Multi-Agent-Systems, Case-Based Reasoning, distributed knowledge sources, knowledge composition

1 Introduction

The knowledge needed to solve a complex problem can not be obtained from only one person. In most cases an expert group ist formed, where each expert has his/her own expertise. By combining the knowledge of all experts in the group, a solution can be better and more reliable. However, not every expert has valuable knowledge for a given problem and only experts who can help solving a problem should be invited to the group.

Our prototype is based on the *SEASALT* architecture [RBA09], which aims at providing a general framework for developing distributed knowledge-based systems. Therefore, the *SEASALT* architecture is domain-independent and modular, which enables the creation of customized systems.

The *Knowledge Line* is part of the knowledge provision within the architecture and handles the knowledge composition that somehow simulates the mechanisms of an expert group for solving complex problems. Having access to many knowledge sources, the Knowledge Engineer designing the *Knowledge Line* has to decide which sources are needed, similar to inviting experts. Because each request can be different, each time a new decision has to be made which knowledge source are necessary to find a solution.

In this paper we are focusing on how distributed knowledge can be combined and whether Case-Based Reasoning (*CBR*) [AP94] enables an efficient selection of knowledge sources for creating combined solutions. We show that a content-based selection of knowledge sources reduces the number of requests significantly and the use of a *Knowledge Map* provides qualitatively good results. The remaining of this paper is structured as follows: Section 2 shows the idea of the *Knowledge Line* and the implementation of the *Multi-Agent-System* [Woo02]. The next section 3 presents the results of the qualitative and quantitative evaluation. In section 4 related approaches are discussed and the final section 5 gives a summary of the paper and provides an outlook on on-going and future work.

1.1 Application Domain: travel medicine

The prototype *docQuery* is an application settled in the travel medicine domain. It can be used to get information about the users destination like prophylactic vaccination for infectious diseases or medication for chronic diseases. Additionally information about security hints, hospitals and possible risks at planned activities are provided. The user submits a query to the application that contains the travel destination, planned arrival and departure, age, activities and chronic diseases. Based on this information, the necessary knowledge sources are identified and requested. The solutions of the individual sources are combined and presented to the user. The presented solution can be used to discuss the necessary arrangements with a doctor. The travel medicine domain has several sub-domains like regions, diseases, medication or activities. Each sub-domain is represented through a separate knowledge source. [BRRSA08, ABD+07]

2 Knowledge Line and Software Agents

The *Knowledge Line* is the central component for knowledge provision in the SEASALT architecture. This component is responsible for the coordination of the access to various heterogeneous knowledge sources and enables the combination of individual solutions. Every knowledge source is a Case-Based Reasoning (*CBR*) system that is managed by a software agent called topic agent. A coordination agent requests the topic agents and combines the solution. The information required by the coordination agent are organized in a so called *Knowledge Map*, which is presented in the following section. A communication agent is responsible for the user's query and to display the solution. At least one *Knowledge Map* agent retrieves the required retrieval paths to find the solution for a given query. Figure 1 shows the structure of the *Knowledge Line* and the software agents the *Knowledge Line* consists of.

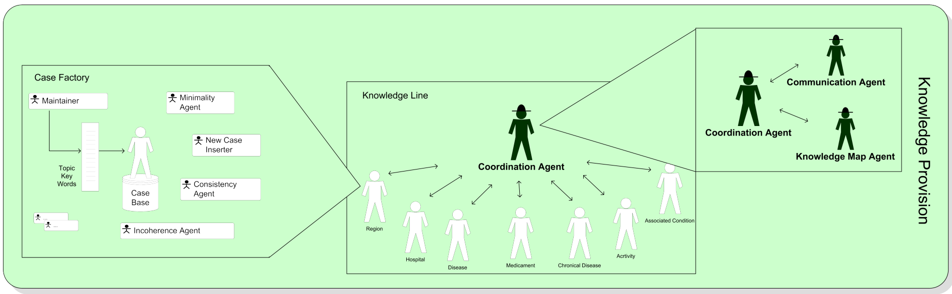

Figure 1: Structure of the Knowledge Line

2.1 Knowledge Map

The basic idea of a *Knowledge Map* comes from Davenport and Prusak [DP00] and was adapted for *Multi-Agent-Systems* from Reichle et al [RBA09, RS08]. Each topic agent is represented as a node of a directed graph and the edges are the dependencies between the agents. The approach from Reichle et al uses a modified Dijkstra algorithm to retrieve the required knowledge sources. For the purpose of the dynamic retrieval, the *Knowledge Map* had to be adapted. One major change is to allow several parallel retrieval paths in the *Knowledge Map*. This allows the system to request topic agents without dependencies to each other in a parallel way. For example the knowledge sources for regions and activities have no dependencies between each other, so the according topic agents can be requested in parallel. Therefore the Dijkstra algorithm is no longer useful, because the algorithm can only retrieve one path in the *Knowledge Map*. Additionally the approach from Reichle et al optimizes the path only using quality criteria like information quality or response time of the software agents. The algorithm can only optimize according to one criterion [RBRSA09] and does not act in knowledge-based way.

Our approach takes the contents of the individual knowledge sources into account to retrieve the required sources. But knowing what sources are available and necessary to get a solution for a certain query is important for the amount of effort. For example, if the user does not want information about activities, then the system can skip the associated knowledge source. In Figure 2 a graphic representation of the default *Knowledge Map* is shown. Our prototype *Multi-Agent-System* has seven knowledge sources and five possible and valid retrieval paths to connect them. These paths are determined by a Knowledge Engineer and if the knowledge sources change, the *Knowledge Map* has to be changed too. To retrieve the required paths a particular *CBR* system with an own software agent is used. The underlying case base contains query characteristics derived from the problem description and information about the required knowledge sources based on the characteristics of the target solution.

There are two major challenges to use a case base to retrieve the necessary knowledge sources. First we need to identify the characteristics of a query, which are capable to determine the needed sources. These characteristics have to be mapped to attribute-value pairs of the case structure. The second challenge is the structure of the solution and the

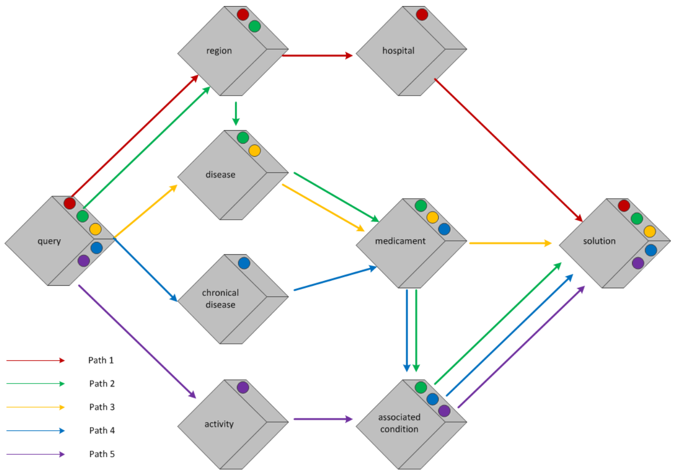

Figure 2: Default *Knowledge Map* with five parallel retrieval paths

information stored. In our prototype the query consists of several parameters like destination, planned activities, age and patient history. Some of these parameters lead directly to an assigned knowledge source and some have an indirect impact on the necessary knowledge. For example, if the parameter destination is set the knowledge source region is directly activated. Based on the destination there are several diseases that could occur and to get the information about the diseases another knowledge source is indirectly required.

Each query consists of seven parameters: destination, journey type, chronical illness, previous diseases, planned activities, age and gender. All these parameters have an impact on the required knowledge sources and therefore an attribute-value pair for every parameter is modeled in the case structure. The parameters destination, chronical illness, previous diseases and planned activties were choosen, because they have an direct impact if the assigned knowledge source is needed. For example, if the user does not specifiy any activities the knowledge source for activities can be ignored. The other three parameters can have an indirekt impact on the required knowledge sources. The age and gender, for example, can have an impact on the planned activities and the risk of some situations like heart attack or diabetes. Additional parameters are possible like existing vaccines or new parameters for additional knowledge sources. For the solution structure we use an approach that applies the sequence of knowledge sources defined in the *Knowledge Map*. This approach assumes a static *Knowledge Map* at runtime. The dependencies between the sources define the possible retrieval paths. Therefore a complete path can be determined from the first knowledge source of a path. This leads to a simple solution structure in the case representation, because only the first nodes of the required retrieval paths have to be stored. With one query on the case base and the information from the *Knowledge Map* all required paths can be found. One possible drawback of this approach is the possibility to retrieve information that is not needed. All knowledge sources of a defined path will be requested even if there are no additional information from the predecessors. For example the path from region over disease and medicament to associated condition, as shown in figure

2, is used. Even if no information about medication for the retrieved diseases is found, the case base for medicament would still be requested. This drawback can be countered with the most complete information.

2.2 Implementation

For the realization of the *Multi-Agent-System* we used the JADE framework [BCPR03] for implementing the heterogenous software agents. To realize the *CBR* systems for the topic agents and the *Knowledge Map* agent we used the open source tool *myCBR*[SRB08]. Not every software agent will be described in detail within this section, but we will give an overview of the implemented agents. Figure 3 shows the architecture of the *Multi-Agent-System* and the communication channels among the agents. A detailed description of the implemented *Multi-Agent-System* can be found in [Reu12].

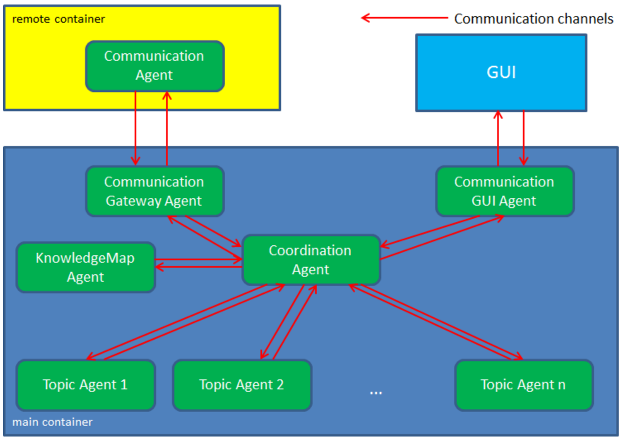

Figure 3: Architecture of the *docQuery* multi-agent system

The *Multi-Agent-System* minimally consists of ten software agents: one communication agent, one coordination agent, one *Knowledge Map* agent and seven topic agents. Since the retrieval of the solution parts has the greatest effort, the system supports several teams of agents consisting of one coordination agent and seven topic agents. In this way we try to avoid a bottleneck when several parallel queries should be processed, because every agent team processes one query.

At the start of the system, the *Knowledge Map* is read and based on the information about the knowledge sources the required topic agents are created, the *CBR* systems started and the case bases imported. The *Knowledge Map* is stored in RDF format and parsed by the main container of the *Multi-Agent-System*. Listing 1 shows an example how information is stored in the *Knowledge Map* for one topic agent. Several parameters can be set for a topic agent:

Listing 1: Example entry for a topic agent in the *Knowledge Map*

```
<rdf:Description rdf:about="docquery-region">
  <conf:agentspercoord>1</conf:agentspercoord>
  <conf:abstract>false</conf:abstract>
  <conf:topic>Region</conf:topic>
  <conf:table>region</conf:table>
  <conf:threshold>1.0</conf:threshold>
  <conf:information quality="100" costs="0" speed="100" limit="0" />
  <conf:link use="Pre_Vaccination_Obl">docquery-disease</conf:link>
  <conf:link use="Pre_Vaccination_Risk">docquery-disease</conf:link>
  <conf:link use="Pre_Vaccination_Std">docquery-disease</conf:link>
  <conf:link use="Pre_Diseases">docquery-disease</conf:link>
  <conf:link use="Name">docquery-hospital</conf:link>
</rdf:Description>
```

- *agentspercoord* sets the number of topic agents created for each coordination agent. This allows us to distribute the load on multiple agents, if a knowledge source is requested very often.
- The parameters *topic* and *table* define the name of the topic agent and the database table where the knowledge is stored. From this table the data is imported into a *CBR*-system.
- *Threshold* sets the minimal similarity of a case to be part of the solution.
- The parameter *link* defines the dependencies between the nodes and sets the attribute which is used to enrich the query.

Additionally, the *Knowledge Map* contains information about the retrieval paths. Several parallel paths can be stored in the *Knowledge Map*. Each path consists of the abstract nodes *query* and *solution* as well as at least two other nodes. In this way the order of requests on the topic agents is defined. Listing 2 shows an example entry for a retrieval path in the *Knowledge Map*. The information about the paths is used by the coordination agent to determine the required retrieval paths based on the set of first nodes retrieved by the *Knowledge Map* agent.

Listing 2: Example entry for a retrieval path in the *Knowledge Map*

```
<rdf:Description rdf:about"retrievalpath">
        <conf:id>1</conf:id>
        <conf:node>docquery-query</conf:node>
        <conf:node>docquery-region</conf:node>
        <conf:node>docquery-hospital</conf:node>
        <conf:node>docquery-solution</conf:node>
</rdf:Description>
```

The coordination agent is the central element in the *Knowledge Line*. It processes the retrieval paths and sends requests to the topic agents. Additionally it combines the solutions from the topic agents to an overall solution. In the prototype this means the coordination agent takes all retrieved cases, eliminates the double ones and sends the remaining cases to the communication agent.

Several challenges had to be solved in the implementation of the parallel processing of the retrieval paths. The coordination agent has to send several parallel requests to different topic agents, according to the necessary paths, and receives the solutions from them. Therefore the agent has to remember which topic agents of a retrieval paths are requested and which is the next topic agent to request. We implemented a data structure to store the progress within a retrieval path. Every time a topic agent sends a solution, the progress of the according path is updated and the coordination agent knows which agent is the next to request. This way we prevent multiple unintentionally requests to a topic agent. Another challenge was handling solutions which contain a large amount of information which shall enrich the next request. An example is the request from the topic agent for the sud-domain region: In many cases, the solution contains several infectious diseases which have to be taken into account. For each disease a request to the according case base is required to receive the information about the medication and prophylaxis. The problem in such a scenario is that multiple requests to the disease topic agent are necessary and the retrieval path is divided into several parallel sub-paths.

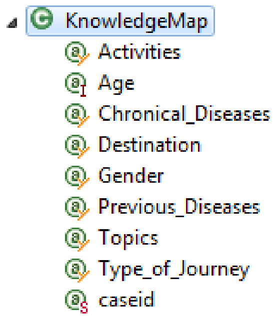

Figure 4: Case structure used by the *CBR*-sytem of the *Knowledge Map* agent

The more multiple agents must be requested, the more sub-paths emerge. Processing additional requests from only one topic agent increases the response time of the *Multi-Agent-System*, creating additional topic agents increases resource consumption. Both approaches have an adverse affect on the system. We are using a different approach. The *Multi-Agent-System* delegates the parallel processing of the sub paths to the *CBR* engine implemented with the Open Source tool *myCBR*. This tool allows to specify multiple values for an attribute. Internally such a request will be handled as follows: all values must be present in a case or for each value an own retrieval is performed [SRB08]. The second option returns multiple matching cases for several diseases in the request. With this approach, only the number of cases that should retrieved has to be adapted. No additional request or a larger number of topic agents is required.

The communication agent receives the request from an user. The request may come from different sources, provided that the data is sent in a compatible format. At the time, requests via a web interface or an iPhone app [Wen12] can be processed. The communication agent accepts and sends data in an XML format. Figure 5 shows a screenshot of the user interface from the iPhone app.

Figure 5: User interface der Iphone app

The *Knowledge Map* agent is responsible for the dynamic determination of the required retrieval paths for a request. For each request a retrieval on the underlying *CBR*-system is performed. The case with the highest similarity over a certain threshold is returned. If no case has a similarity over the threshold, no case is returned and the coordination agent uses the standard paths to find a solution. The case representation consists of seven attributes to describe the situation and one attribute for the solution. Figure 4 shows the case structure used by the *CBR*-system of the *Knowledge Map* agent. The solution attribute contains the first nodes of the reuqired retrieval paths. For each attribute, it was necessary to model the knowledge, that a parameter is not set. In determining the required retrieval paths, it is important to distinguish whether the knowledge of a sub-domain is required or not. In addition, the absence of a parameter is more important than a concrete value. The following request is used to illustrate the problem. A query Q with the values *Indonesia* for the attribute region and *Rafting* for the attribute activity. How to accept, there exist two cases C_1 and C_2 with the following values: C_1 contains *Indonesia* and *NoActivity*, C_2 contains *Malaysia* and *Hiking*. Using a similarity modeling with equal weighting, C_1 would have a higher similarity than C_2, because C_1 has a match, while C_2 does not match. But, this would lead to a wrong solution because the retrieval path for the activities is missing. The knowledge about the planned activities would be missing in the overall solution. The solution of C_2 would be correct, because the solution contains both required paths, even if the specific values do not match. To solve this problem, the attributes destination, activities, previous diseases and chronic illness, uses a taxonomy for modeling the local similarity. Each taxonomy contains on one branch the value for the absence of a parameter and on the other branch the specific values of the attribute. In this way a minimum similarity is assigned to the specific values between each other, while they have similarity of zero to the value for the absence of a parameter.

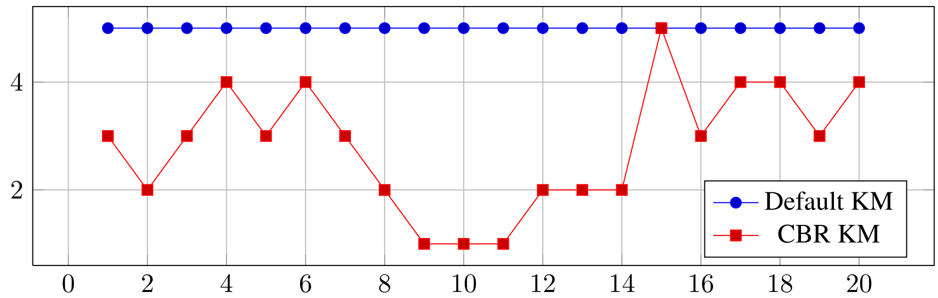

Figure 6: Number of paths used in query series 1

3 Experimental Results

To evaluate the efficiency of the prototype we focus on the selection of the retrieval paths and the content of the retrieved solution. Our hypotheses are that using a content-based selection of the knowledge sources reduces the number of required request to find a solution and using a *Knowledge Map* provides a complete and correct solution. To prove the hypothesis we used the implemented *Knowledge Map* agent for content-based path selection. Then an empirical approach was used to adress the quantitative and qualitative aspects. 20 queries were build and send twice to the *Multi-Agent-System*. The queries were choosen so that they cover all possible combinations of the defined retrieval paths. In this way, it can be checked wheter the selected retrieval paths match the query. At the first round, the 20 queries are answered with the use of the *Knowledge Map* agent. At the second round the queries are answered without the use of the *Knowledge Map* agent. The retrieval paths used in both rounds are compared to get a statement about the completeness of the used retrieval paths. The used paths of the first series are complete if the solution contains the same information as the solution retrieved with standard paths. If the solution is complete with less retrieval paths then the standard paths, the first hypothesis (Requiring less requests) is proved. For the evaluation of the content of the retrieved solution an expert is used. The expert gets the solution of the queries from the first series and rates these solutions with regards to completeness and correctness. If the experts rate the content of more than 15 solutions as complete and correct, our second hypothesis (Using a *Knowledge Map* provides complete and correct solutions) is correct.

In the following the results of the evaluation are presented and discussed. Figure 6 shows the used retrieval paths for the 20 queries in both series. In the table the number of paths for every query is listed. Without the *Knowledge Map* agent every time the five standard paths are used. The diagram shows the average number of paths used to get a solution. With the *Knowledge Map* agent it took only half the paths to get the same solution. Less used paths results in less sources to request. In particular in the use of paid oder complex computing services to get knowledge, a minimum number of requests is desirable. Additionally the retrieved solution parts are compared between the to series to check if the retrieved parts are the same. In figure 7 the numbers of concept for each solution part are shown with use of the *Knowledge Map* agent. In both series the numbers are mostly the

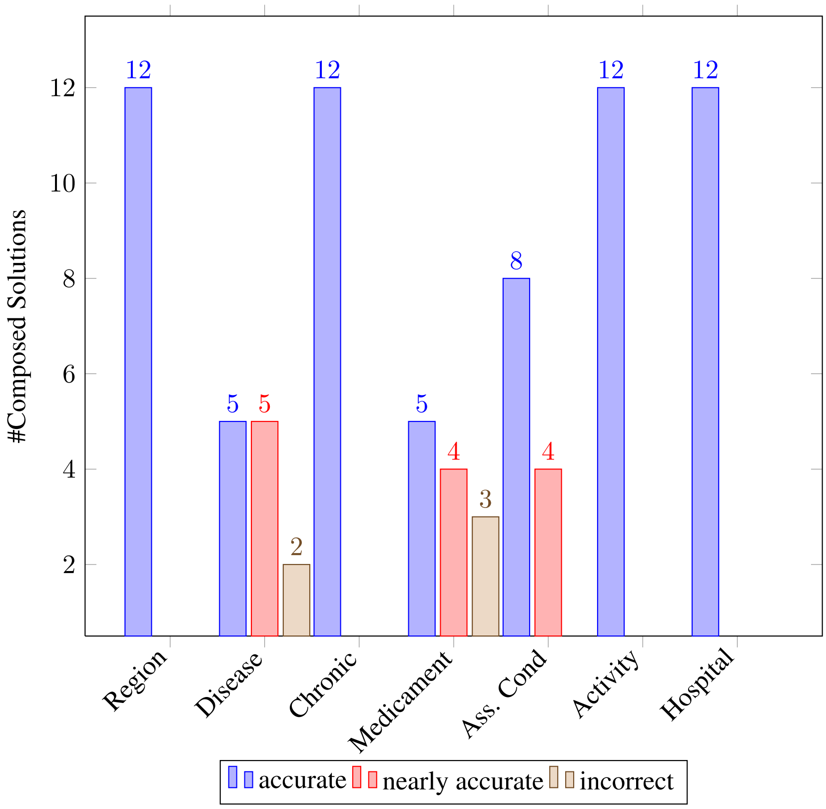

Figure 7: Classification of retrieved solutions

same. Only for query 9 to 16 the numbers differ, because in this queries no destination is set. In the first series the associated knowledge source is not used, but in the second series the standard paths requested the sources anyway. Based on these results we can say, that the *Knowledge Map* agent is capable of retrieving only the required paths and boosts the efficiency and competence of the prototype. Using the standard paths sometimes unnecessary informations are retrieved.

The experts used to rate the content of the solutions are from *mediScon*[1]. This company is an information and counseling service for travel medicine. The experts rates all queries from the first series with a destination set. Queries 9 to 16 were not rated because without a destination a reasonable rating is not possible. Figure 7 shows the classification of the retrieved solutions. The chart shows that most solutions are accurate. This means all expected concepts are retrieved. Some solutions are nearly accurate, what means one conecpt was missing or incorrect. For example a disease was not retrieved or a wrong medication. If more than one concept was missing or incorrect the solution was classified as incorrect.

[1] http://www.mediscon.de

Based on this results, the solutions of the *Multi-Agent-System* are useful to the respective request. But some points can still be improved. However, these improvements relate to the modeling of the case structure and the available data. The interaction of the individual agents in the context of the knowledge line and the use of the *Knowledge Map* agent works as desired. The hypothesis, that a content-based selection of knowledge sources reduces the number of request is confirmed.

4 Discussion of Related Work

Combining distriubted knowledge sources to a qualitative solution can be done in different ways. Approaches and systems that adresses this problem are AMAL [OP07], EpiSpider [TKF$^+$07] and the Dempster-Shafer theory [Sha90]. While AMAL uses software agents with *CBR* systems that argue with each other to find a solution, EpiSpider uses information retrieval technologies and natural language processing to find requires knowledge sources. The Dempster-Shafer theory uses a different approach and calculates probabilities for the veracity of each partial solution and from this the probability for the accuracy of the overall solution. Our system uses software agents like AMAL to find a high-quality solution. However in our prototype the *Knowledge Map* agent is responsible for choosing the required knowledge sources. In this way a pre-selection of the knowledge source is taken. In addition, the coordination agent assumes the task to combine the partial solutions and thus sort out unneccessary knowledge. At AMAL all agents can bring their knowledge, even if it is unneccessary.

Multi-Agent-Systems allow a load distribution and delegation of tasks to different agents. Our prototype uses this by spreading knowledge on different topic agents. As a result, it is possible to request the individual *CBR* systems in a parallel way. It is also possible to increase the number of agents, so that a large number of requests can be processed at the same time. A different approach to load balancing uses the framework Hadoop citeShaferetal2010, Shavachko2010. It relies on the MapReduce engine to distribute tasks to different servers. While Hadoop is very performant, the maintenance of the knowledge is expensive. *CBR* systems, such as in our approach, have benefits at modularization and maintenance of the knowledge.

5 Conclusion and Outlook

In this paper we presented a *Multi-Agent-System* that is able to select the required knowledge source to solve a given problem. Based on the *SEASALT*architecture we developed a concept for the *Knowledge Line* and strategies for determining the necessary knowledge sources to solve a problem. We described the structure of the *Knowledge Map* and how it is used by the *Multi-Agent-System* to retrieve the required retrieval paths. As the results section shows the prototype is able to provide high-quality solutions. Using a *Knowledge Map* agent to determine the required retrieval paths indicates that fewer requests to find a solution are necessary, as in the processing of all knowledge sources. Because the solutions with use of the *Knowledge Map* agent contains the same information as without it, the use of the *Knowledge Map* agent leads clearly to an increase in the performance and the competence of the *Multi-Agent-System*. In this way, it could be shown that the approach of the selection in terms of content of knowledge sources works.

The successful implementation of the *Knowledge Line* helps finding new ways to apply *knowledge-based system*s in practice. However the rating and the comments of the experts from *mediScon* have also shown that more knowledge is required to lead to better solutions. The acquisition and deployment of knowledge certainly represents a challenge and can be identified in this context as so-called knowledge aquisition bottleneck. The initial effort to integrate the knowledge of experts in the system is high especially unter the aspect that travel medicine is a very complex domain. For each heterogenous subdomain an expert is required. Although still a lot can be improved, this implementation provides a basis for a practical application of *knowledge-based system*s in travel medicine.

References

[ABD$^+$07] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Regis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative Multi-Expert-Systems – Realizing Knowledge-Product-Lines with Case Factories and Distributed Learning Systems. In Joachim Baumeister and Dietmar Seipel, editors, *KESE @ KI 2007*, Osnabrück, September 2007.

[AP94] Agnar Aamodt and Enrice Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. *AI Communications*, 7(1):39–59, 1994.

[BCPR03] Fabio Bellifemine, Giovanni Caire, Agostino Poggi, and Giovanni Rimassa. JADE A White Paper. Technical report, Telecom Italia Lab, 2003.

[BRRSA08] Kerstin Bach, Meike Reichle, Alexander Reichle-Schmehl, and Klaus-Dieter Althoff. Implementing a Coordination Agent for Modularised Case Bases. In *Proceedings of the 13th UK Workschop on Case-Based Reasoning*, pages 1–12, 2008.

[DP00] Thomas H. Davenport and Laurence Prusak. *Working Knowledge: How Organizations Manage What they Know*. Havard Business School Press, 2000.

[OP07] Santi Ontanon and Enric Plaza. An Argumentation-Based Framework for Deliberation in Multi-agent Systems. In *ArgMAS 07 Proceedings of the 4th International Conference on Argumentation*, pages 178–196, 2007.

[RBA09] Meike Reichle, Kerstin Bach, and Klaus-Dieter Althoff. The SEASALT Architecture and its Realization within the docQuery Project. In *Proceedings of the 32nd Annual Conference on Artifical Intelligence*, pages 556–563, 2009.

[RBRSA09] Meike Reichle, Kerstin Bach, Alexander Reichle-Schmehl, and Klaus-Dieter Althoff. Management of Distributed Knowledge Sources for Complex Application Domains. In *Proceedings of the 5th Conference on Professional Knowledge Manegement*, pages 128–138, 2009.

[Reu12] Pascal Reuss. Konzept und Implementierung einer Knowledge Line - Retrievalstrategien für modularisierte, homogene Topic-Agenten innerhalb eines Multi-Agenten-Systems. Master's thesis, University of Hildesheim, 2012.

[RS08] Alexander Reichle-Schmehl. Entwurf und Implementierung eines Softwareagenten zur Koordination des dynamischen Retrievals auf verteilten, heterogenen Fallbasen. Bachelor-arbeit, Universität Hildesheim, 2008.

[Sha90] Glenn Shafer. Perspectives on the theory and practice of belief functions. *International Journal of Approximate Reasoning*, 3:1–40, 1990.

[SRB08] Armin Stahl and Thomas Roth-Berghofer. Rapid Prototyping of CBR Applications with the Open Source Tool myCBR. In *Advances in Case-Based Reasoning, Lecture Notes in Computer Science*, pages 615–629, 2008.

[TKF+07] Herman Tolentino, Raul Kamadjeu, Paul Fontelo, Fang Liu, Michael Matters, Marjorie Pollack, and Larry Madoff. Scanning the Emerging Infectious Diseases Horizon. *Advances in Disease Surveillance*, 2:169, 2007.

[Wen12] Carsten Wenzel. Userinterface für DocQuery auf mobilen Endgeräten. Master's thesis, Universität Hildesheim, 2012.

[Woo02] Michael Wooldridge. *An Introduction to Multiagent Systems.* John Wiley & Sons Ltd., West Sussex, England, 2002.