

Management of Distributed Knowledge Sources for Complex Application Domains

Meike Reichle, Kerstin Bach, Alexander Reichle-Schmehl, and Klaus-Dieter Althoff

Intelligent Information Systems Lab

University of Hildesheim

Marienburger Platz 22

31141 Hildesheim, Germany

{reichle, bach, reichle-schmehl, althoff}@iis.uni-hildesheim.de

Abstract: This paper focuses on knowledge management for complex application domains using Collaborative Multi-Expert-Systems. We explain how different knowledge sources can be described and organised in order to be used in collaborative knowledge-based systems. We present the docQuery system and the application domain travel medicine to exemplify the knowledge modularisation and how the distributed knowledge sources can be dynamically accessed. Further on we present a set of properties for the classification of knowledge sources and in which way these properties can be assessed.

1 Introduction

Today's knowledge-based systems have to deal with increasingly complex application domains. One way of dealing with this increasing complexity are distributed systems such as multi-agent systems [Wei99]. However, in order to realise a truly distributed knowledge-based system not only the knowledge processing step has to be carried out in a distributed way but also the knowledge acquisition step. Service-oriented architectures [PTDL03] are one example for the use of distributed information sources but they mostly focus on services that are closer to the processing of knowledge than its acquisition. The Collaborative Multi-Expert Systems approach [ABD+07] addresses the challenge of distributed knowledge acquisition in its first instantiation, the SEASALT architecture [BRA07]. This paper's focus lies on SEASALT's distributed knowledge sources and their management and (optimised) querying using a Coordination Agent [BRRSA08]. The Coordination Agent builds a graph representation of all available knowledge sources, the graph's shape is based on the knowledge sources' respective input/output dependencies [RS08]. Based on this graph an optimal route through the graph – that is an optimal combination of the information offered by the respective knowledge sources – is computed. In order to optimise this route and thus the resulting information the knowledge sources have to be classified using appropriate properties. This paper evaluates and presents possible properties that can be used to model and describe heterogeneous knowledge sources of all sorts.

Section 2 presents the application domain travel medicine, which is the application domain of the docQuery project [Bac07], the project within which the presented work has been developed. Section 3 presents the underlying concept of knowledge modularisation and the Coordination Agent. Subsection 3.1 presents the graph-based representation, the

Knowledge Map, in detail and subsection 3.2 gives a detailed description of the computation of retrieval routes based on the Knowledge Map. Section 4 presents the knowledge source properties we have identified, their possible values and possibilities for their automated assessment. Section 5 presents related work in the areas of knowledge modularisation and the description of knowledge sources. The paper closes with a conclusion and outlook in section 6.

2 Application Domain: Travel Medicine

During the last decade travelling to different places, experiencing new cultures and meeting new people all over the world has become more and more popular. In preparation for a healthy journey it is important to get high quality and reliable information on travel medicine prevention. Travel medicine is the specialised area of medicine that deals with medical issues like diseases, vaccinations, etc., which might occur before, during and after a journey. In fact, it focuses on what happens to people when they change their regular environment, for example when travelling by car, train or airplane to different places.

There are already many websites and web forums in which travel medicine information can be found (e.g., which vaccinations should be administered when someone plans to travel to a given country). The main drawback of such websites and web forums is that they usually do not contain all necessary medical information and the traveller has to visit many pages to receive all the information he needs. Thus, it is a difficult and time-consuming task to gather all the information for a travel destination. Furthermore, the editors of the sites are mostly unknown and travellers cannot evaluate whether the given information is trustworthy, complete and/or correct.

We aim to remedy these problems and will create docQuery, an intelligent information system on travel medicine in co-operation with a team of certified doctors of medicine with a strong background in travel related medicine. This offers us the possibility to establish a community for experts in which they can exchange their knowledge on their expertise (e.g., coping with chronic illnesses during a journey) and get new information from their colleagues.

Based on our SEASALT architecture, we will implement the docQuery² application [Bac07] which will provide the travellers with travel medicine information tailored to suit their journey. Queries to the system contain data like travel period, destination, age of the traveller. We will be able, by the means of docQuery, to create tailored information leaflets that cover all aspects a consultation at a travel medicine expert would contain. The knowledge within docQuery is modularised with knowledge sources covering topics like regions, diseases, medicaments, activities, chronic illnesses, etc. Following

² docQuery is a project in co-operation with mediScon worldwide (and TEMOS, the telemedicine project of the Institute of Aerospace Medicine at the German Aerospace Centre - DLR

the SEASALT architecture the knowledge sources contain information extracted from community knowledge as well as human domain experts.

3 Knowledge Modularisation

Following our approach of Collaborative Multi-Expert Systems the knowledge sources, which are used to store and provide knowledge, are mostly distributed. When dealing with complex application domains it is easier to maintain a number of heterogeneous knowledge sources than one monolithic knowledge source. The knowledge modularisation within SEASALT is organised in the Knowledge Line that is based on the principle of product lines as it is known from software engineering [vSR07] and we apply it to the knowledge in knowledge-based systems, thus splitting rather complex knowledge in smaller, reusable units (knowledge sources). Moreover, the knowledge sources contain different kinds of information as well as there can also be multiple knowledge sources for the same purpose. Therefore each source has to be described in order to be integrated in a retrieval process which uses a various number of knowledge sources.

The approach presented in this work does not aim at distributing knowledge for performance reasons, instead we are planning to specifically extract information for the respective knowledge sources from internet communities or to have experts maintaining one knowledge base. Hence, we are creating knowledge sources, especially Case-Based Reasoning systems, that are accessed dynamically according to the utility and accessibility to answer a given question. Each retrieval result of a query is a part of the combined information as it is described in the CoMES approach [ARB+07].

3.1 Knowledge Map

The Knowledge Map organises all available knowledge sources that can be accessed by a so-called Coordination Agent that creates individual requests and combines information. The term Knowledge Map originates in Davenport's and Prusak's work on Working Knowledge [DP00] in which they describe a knowledge map from the organisational point of view mapping human experts in order to ensure that everybody in a company knows who is an expert in a certain domain. We transfer this concept in an intelligent agent framework that coordinates different knowledge sources.

A Knowledge Map KM consists of a number of Topic Agents TA that are depending on each other and each consist of a software agent A on top of a knowledge base KB. Thus it can be defined as follows:

$$KM = fTA1; TA2; TA3; \dots; TA_n \text{ with } TA = (KB;A) \quad (1)$$

A Topic Agent is a knowledge-based system itself and the software agent queries it. The Topic Agent collaborates with the Coordination Agent that navigates through the Knowledge Map and asks subsequent questions to the individual Topic agents thus creating an individual path through the map. There are dependencies $Dep_{constraint}$ between the Topic Agents which define that sequence and influence the retrieval executed by one of

the subsequent Topic Agents. A dependency exists if one agent’s output serves as another agent’s input and thus enforces a subsequent query. Since the dependencies between Topic Agents can take any form, we decided to implement the Knowledge Map as a graph where each Topic Agent is represented by a node and directed edges denote the dependencies.

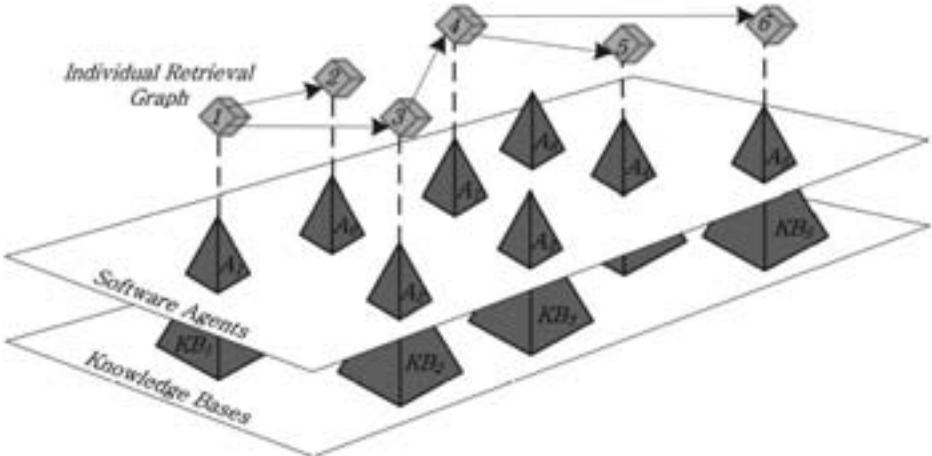


Figure 1: Knowledge Map containing Topic Agents and Knowledge Bases

Figure 1 shows a Knowledge Map containing the knowledge bases and software agents as well as an example for a possible path through the knowledge sources.

3.2 Computing Retrieval Graphs

Retrieval paths are computed based on the information a user gives in an individual query and the properties of the knowledge sources.

Our implementation covers an a-priori computation of the retrieval path, the Knowledge Map itself is stored as an XML document. We use RDF as the wrapper format and describe the individual nodes with a namespace of our own. More details concerning the XMLFormat can be found in [RS08]. Based on the knowledge map we then use a modified Dijkstra algorithm [Dij59] to determine an optimal route over the graph. If confronted with redundant knowledge sources the algorithm tries to optimise the path according to the knowledge sources respective properties. To that end the algorithm is modified in such a way that it optimises its route by trying to maximise the arithmetic mean of all queried nodes. In the case of a tie between two possible routes the one with the lesser variance is chosen.

4 Classification of Knowledge Sources

As described in the previous section a set of properties is assigned to the knowledge sources. These properties do not only describe the individual sources but are also used for optimising the query path. When working with distributed and – most importantly – external sources it is of high importance to be able to assess, store and utilise their characteristics in order to achieve optimal retrieval results.

4.1 Knowledge Source Properties

Considering knowledge sources, different characteristics, and aspects on which to assess knowledge source properties come to mind. The possible properties can refer to content (e.g. quality or topicality) as well as meta-information (e.g. answer speed or access limits). In detail we have identified the following knowledge source (meta and content) properties:

Meta properties are:

- **Access Limits:** Some services, for instance some Google Maps³ services or *Projekt Deutscher Wortschatz*⁴, only allow a certain number of requests per time unit. These limits can make a service less attractive and have to be observed in order to avoid being blocked.
- **Answer Speed:** The knowledge source's reaction time.
- **Economic Cost:** While most information sources are free to use, there are also many commercial ones that can be used, but should also be avoidable.
- **Language:** The query language, for instance SQL, RMI or simple query terms.
- **Format:** The results' format, for instance XML, HTML, data base tables, or pure text.
- **Structure:** How structured the results are: possible values are structured (e.g. tables), semi-structured (e.g. RSS), and unstructured (e.g. text).
- **Cardinality:** The amount of results, for instance a single one or a set of cases or tuples.
- **Trust or Provenance:** Not all knowledge sources are equally trustworthy and reliable; this has to be reflected in their properties.⁵

³ <http://maps.google.com>

⁴ <http://wortschatz.uni-leipzig.de/>

⁵ It is arguable if this is exclusively a meta property since provenance also affects content but we have decided to count it among the meta properties.

Content properties:

- **Content:** What knowledge the knowledge source actually offers. Content can be semantically described using a semantic description language such as RDF (Resource Description Framework [BM04]).
- **Expiry:** Some kinds of knowledge or information are more time dependent than others. A geographic knowledge source providing coordinates of places can easily be cached for a long time, stock quotations on the other hand cannot.
- **Up-to-dateness:** If the offered information is time-dependent, how up-to-date are the results?
- **Coverage:** How good is the knowledge source's topic covered? For instance in a service providing stock quotations, does the source offer quotations for every issued stock or only on selected ones?
- **Completeness:** How complete is the information offered, here the question is not how complete a topic is covered but how complete the provided information is.

The distinction between coverage and completeness is most easily illustrated in an example: Given a need for geographic information the free encyclopaedia Wikipedia would be a knowledge source with a high coverage, since it lists not only countries but also provinces, regions, cities, and villages. The CIA World Factbook⁶ on the other hand has a much lower coverage since it only offers information on a country-basis. With regard to completeness the World Factbook would have a high value since it offers a full set of information for every included country, while in the Wikipedia there are also so called stubs, articles that only contain of one or a few sentences and thus are very incomplete.

While the properties presented above can be easily described and modelled, there are also more complex knowledge source properties. One of these more complex properties is quality: The quality of a knowledge source comprises many different aspects and we thus propose to also allow for compound properties to also permit the description of complex properties. Compound properties are the (weighted) sum of any number of the above presented simple topics.

4.2 Assessment of Knowledge Source Properties

Not all of the properties presented above are fully unrelated. The properties language, format, structure and cardinality for instance are partially related which allows for some basic sanity checks of their assigned values; also some of the properties such as answer

⁶ <https://www.cia.gov/library/publications/the-world-factbook/>

speed, language or structure can be automatically assessed. Apart from these possibilities for automation the knowledge source properties currently have to be assessed and maintained manually by a Knowledge Engineer who assigns values to the properties and

Property	Type	Description
Economic Cost	<i>Float</i>	Cost per query
Access Limits	<i>Integer</i>	Number of queries per minute
Answer Speed	<i>Integer</i>	Number of milliseconds
Language	<i>SetOfValues</i>	E.g. XML, SQL, Text, ...
Format	<i>SetOfValues</i>	E.g. RSS, XML, HTML, DB tuples, Text, ...
Structure	<i>SetOfValues</i>	E.g. Structured, Semi-Structured, Unstructured
Cardinality	<i>SetOfValues</i>	E.g. Single, Set
Content	<i>SetOfValues</i>	Semantic description of the source's content
Provenance	<i>IntRange</i>	A value in a specified range
Expiry	<i>IntRange</i>	A value in a specified range
Up-to-dateness	<i>IntRange</i>	A value in a specified range
Coverage	<i>IntRange</i>	A value in a specified range
Completeness	<i>IntRange</i>	A value in a specified range

Table 1: Knowledge source properties and their possible values

keeps them up to date. Adapting the properties' values based on feedback is only partially possible since feedback is mostly given on the final, combined result and it is thus difficult to propagate back to the respective knowledge sources. Also the more differentiated feedback is needed (in order to be mapped to the respective properties) the less feedback is given, so a good balance has to be found in this regard. Despite these difficulties the inclusion of feedback should not be ruled out completely. Even if good knowledge sources are affected by bad general feedback and the other way around the mean feedback should still provide a basic assessment of a knowledge source's content and can for instance be included in a combined quality measure. Depending on the respective properties we have defined possible values for all of them, table 1 illustrates all properties and their possible values. Obviously, not all properties are usable for routing optimisation. There are some properties like format, language, structure or content that cannot be used in the routing process since no valency can be assigned to them, that is one possible value cannot be judged as better or worse as the other. The computation of the routes with regard to defined properties is carried as described in section 3.2.

5 Discussion of Related Work

The approach of distributed sources has been a research topic in Information Retrieval since the mid-nineties. An example is the Carrot II project [CKM+02], which also uses a multi-agent-system to co-ordinate the document sources. However, most of our knowl-

edge sources are CBR-systems, which is the reason why we concentrate on CBR-approaches.

The issue of differentiating case bases in order to be more suitable for its application domain has been discussed before. Weber et al. [WGM08] introduce the horizontal case representation, a two case base approach in which one contains the problem and the other one the solutions. They motivate splitting up the case bases for a more precise case representation, vocabulary and a simplified knowledge acquisition.

Retrieval strategies have been discussed in the context of Multi-Case-Base Reasoning in [LS02]. Leake and Sooriamurthi explain how distributed cases can be retrieved, ranked and adapted. Although they are dealing with the same type of case representations of the distributed case bases, both approaches have to determine whether a solution or part of solution is selected or not. The strategy of Multi-Case-Base Reasoning is to either dispatch cases if a case-base cannot provide a suitable solution or to use cases of more than one case base and initiate an adaptation process in order to create one solution.

Collaborating case bases have been introduced by Ontañón and Plaza [OP01] who use a multi-agent system to provide a reliable solution. The multi-agent system focuses on learning which case base provides the best results, but they do not combine or adapt solutions of different case bases. Instead their approach focuses on the automatic detection of the best knowledge source for a certain question.

Combining parts of cases in order to adapt given solutions to a new problem has been introduced by Redmond in [Red90] in which he describes how snippets of different cases can be retrieved and merged into other cases, but in comparison to our approach, Redmond uses similar case representations from which he extracts parts of cases in order to combine them. His approach and the knowledge provision in SEASALT have in common that both deal with information snippets and put them together in order to have a valid solution. Further on, Redmond mostly concentrates on adaptation while we combine information based on retrieval and routing strategy.

Our notion of knowledge source properties is comparable to and thus benefits from advances in the respective field in CBR like the recent work of Briggs and Smyth [BS08], who also assign properties, but to individual cases. On the other hand the graph-like representation of the knowledge sources and its use in the composition of the final results do not have a direct equivalent in CBR. It depends on the cases' separation by topic and a clear dependency structure of the topics (*e.g. the country determines the possible diseases, the diseases determine the respective vaccinations and precautions, etc.*) which is not necessarily given in traditional CBR.

6 Conclusion and Outlook

In this paper we presented how knowledge management for complex application domains can be realised using Collaborative Multi-Expert-Systems. We introduced doc-Query (as our application domain) which is partially based on the results presented in this work. We explained the knowledge modularisation and how the distributed knowl-

edge sources can be dynamically accessed defined by retrieval paths based on Knowledge Maps. Further on, we explained how knowledge sources can be classified using a set of properties and in which way these properties can be assessed.

Currently, our implementation covers an a-priori computation of the retrieval path and we are planning to extend the computation toward a more flexible and subsequent, result-dependent routing. We plan to extend our algorithm by weighting and combining the considered properties to represent their relevance in the overall solution. Further on, we will evaluate the automated integration of feedback about knowledge sources (e.g. regarding the quality of the result or the response time) in the computation of the retrieval paths. The computation of retrieval graphs can be improved by implementing a more flexible computation depending on the subsequent results by adjusting the retrieval path during retrieval time.

Because of the fact, that we use a number of different knowledge sources we consider using results of the explanation-based research [RB04] to describe to the user on which sources the answer depends in order to have more trustworthy solutions. Further on, these explanations will help use to evaluate our knowledge sources aiming at a more precise, user-comprehensible and transparent system.

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, *Accepted for the Proceedings of the 3rd Workshop on Knowledge Engineering and Software Engineering (KESE 2007)*, Osnabrück, September 2007.
- [ARB+07] Klaus-Dieter Althoff, Meike Reichle, Kerstin Bach, Alexandre Hanft, and Regis Newo. Agent Based Maintenance for Modularised Case Bases in Collaborative Multi-Expert Systems. In *Proceedings of AI2007, 12th UK Workshop on Case-Based Reasoning*, pages 7–18, dec 2007.
- [Bac07] Kerstin Bach. docQuery - A Medical Information System for Travellers. Internal project report, 2007.
- [BM04] Dave Beckett and Brian McBride. RDF/XML Syntax Specification (Revised). Technical report, World Wide Web Consortium, February 2004. <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [BRA07] Kerstin Bach, Meike Reichle, and Klaus-Dieter Althoff. A Domain Independent System Architecture for Sharing Experience. In *Proceedings of LWA 2007, Workshop Wissens- und Erfahrungsmanagement*, pages 296–303, September 2007.

- [BRRSA08] Kerstin Bach, Meike Reichle, Alexander Reichle-Schmehl, and Klaus-Dieter Althoff. Implementing a Coordination Agent for Modularised Case Bases. In Miltos Petridis, editor, *AI 2008, 13th UK Workshop on Case-Based Reasoning*, pages 1–12, December 2008.
- [BS08] Peter Briggs and Barry Smyth. Provenance, Trust, and Sharing in Peer-to-Peer Case-Based Web Search. In Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft, editors, *Advances in Case-Based Reasoning, Proceedings of the 9th European Conference, ECCBR 2008*, volume 5239 of *Lecture Notes in Computer Science*, pages 89–103. Springer, 2008.
- [CKM+02] R. Scott Cost, Srikanth Kallurkar, Hemali Majithia, Charles Nicholas, and Yongmei Shi I. Integrating distributed information sources with CARROT II. In Matthias Klusch, Sascha Ossowski, and Onn Shehory, editors, *Cooperative Information Agents VI, 6th International Workshop, CIA 2002, Madrid, Spain, September 18-20, 2002, Proceedings*, volume 2446 of *Lecture Notes in Computer Science*. Springer, 2002.
- [Dij59] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [DP00] Thomas H. Davenport and Laurence Prusak. *Working Knowledge: How Organizations Manage What they Know*. Harvard Business School Press, May 2000.
- [LS02] David B. Leake and Raja Sooriamurthi. Automatically Selecting Strategies for Multi-Case-Base Reasoning. In *ECCBR '02: Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, pages 204–233, London, UK, 2002. Springer-Verlag.
- [OP01] Santiago Ontañón and Enric Plaza. Learning When to Collaborate among Learning Agents. In Luc De Raedt and Peter Falch, editors, *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, volume 2167, pages 394–405, London, UK, 2001. Springer-Verlag.
- [PTDL03] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. Service-oriented computing. *Communications of the ACM*, 46:25–28, 2003.
- [RB04] Thomas R. Roth-Berghofer. Explanations and Case-Based Reasoning: Foundational Issues. In Peter Funk and Pedro A. Gonz’alez Calero, editors, *Advances in Case-Based Reasoning*, pages 389–403. Springer-Verlag, September 2004.
- [Red90] Michael Redmond. Distributed Cases for Case-Based Reasoning: Facilitating Use of Multiple Cases. In *AAAI*, pages 304–309, 1990.
- [RS08] Alexander Reichle-Schmehl. Entwurf und Implementierung eines Softwareagenten zur Koordination des dynamischen Retrievals auf verteilten, heterogenen Fallbasen. Master’s thesis, Hildesheim University, September 2008.

- [vSR07] Frank van der Linden, Klaus Schmid, and Eelco Rommes. *Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering*. Springer, Berlin, Heidelberg, Paris, 2007.
- [Wei99] Gerhard Weiß. *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.
- [WGM08] Rosina Weber, Sidath Gunawardena, and Craig MacDonald. Horizontal Case Representation. In Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft, editors, *Advances in Case-Based Reasoning, Proceedings of the 9th European Conference, ECCBR 2008*, volume 5239 of *Lecture Notes in Computer Science*, pages 548–561. Springer, 2008.