The SEASALT Architecture and Its Realization within the docQuery Project

Meike Reichle, Kerstin Bach, and Klaus-Dieter Althoff

Intelligent Information Systems Lab
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
{reichle,bach,althoff}@iis.uni-hildesheim.de

Abstract. SEASALT (Sharing Experience using an Agent-based System Architecture LayouT) presents an instantiation of the Collaborating Multi-Expert Systems (CoMES) approach [1]. It offers an application-independent architecture that features knowledge acquisition from a web-community, knowledge modularization, and agent-based knowledge maintenance. The paper introduces an application domain which applies SEASALT and describes each part of the novel architecture for extracting, analyzing, sharing and providing community experiences in an individualized way.

1 Introduction

The development and expansion of Web 2.0 applications in the last years has resulted in the fact that formalized and structured documents have been largely replaced by individually structured and designed documents and experiences. Instead of using ready-made forms or templates to express their opinions, Web 2.0 participants present their experiences and ideas individually - for example via blog or forum posts, on mailing lists or in wikis. In order to keep up with the development towards more sophisticated social software applications, the techniques and approaches for intelligent information systems have to develop further as well. Traditional approaches like strictly structured monolithic data bases or highly specialized Text Mining approaches cannot deal sufficiently with the wealth of experiences provided in todays Word Wide Web.

In this paper we present a novel architecture for extracting, analyzing, sharing and providing community experiences. Our architecture is geared to real world scenarios where certain people are experts in special domains and the knowledge of more than one expert as well as the composition of a combined solution are required in order to solve a complex problem.

The core methodology for the realization of SEASALT (Sharing Experience using an Agent-based System Architecture LayouT), is Case-Based Reasoning (CBR) [2]. CBR has already been successfully applied in many industrial and academical applications [3,4]. Moreover, CBR is a technology for reusing experiences [5] and the technologies used within a CBR system can be customized according to a given domain.

B. Mertsching, M. Hund, and Z. Aziz (Eds.): KI 2009, LNAI 5803, pp. 556–563, 2009.

2 docQuery: An Application Based on SEASALT

Travel medicine is the prevention, management and research of health problems associated with travel play a major role alongside with individual aspects concerning the health status of the traveler and the desired destination. Therefore, information about the a traveler's home region as well as the destination region, the activities planned and additional conditions have to be considered when giving medical advice. Travel medicine starts when a person moves from one place to another by any kind of transportation and ends after returning home healthy. In case a traveler gets sick after a journey a travel medicine consultation might also be required.

Nowadays it is easier than ever to travel to different places, experience new cultures and get to know new people. In preparation for a healthy journey it is important to get a high quality and reliable answer on travel medicine issues. Both laymen and experts should get information they need and, in particular, they understand. For that reason we would like to introduce docQuery - a medical information system for travelers. Whether somebody travels frequently or occasionally, on business or for leisure, individually or with the whole family, docQuery should be able to provide individualized knowledge. The docQuery project focuses on high quality information that can be understood by everybody and maintained by a number of travel medicine experts, supported by intelligent methods executed by agents. Furthermore, the various and heterogeneous fields require independently organized knowledge sources. In comparison to traditional approaches that mostly rely on one monolithic knowledge resource, the docQuery system will adapt to the organization of knowledge given by the expert. An analysis of the expert's tasks shows that the information gathered from different channels (mailing lists, web forums, literature) has to be organized, analyzed, and synthesized before it can be provided. docQuery concentrates on a web community in which experts exchange and provide qualified information. docQuery can be used by inserting the key data on a travelers journey (like travel period, destination, age(s) of traveler(s), activities, etc.) and the system will prepare an individual composed information leaflet right away. The traveler can take the information leaflet to a general practitioner to discuss the planned journey. The leaflet will contain all the information needed to be prepared and provide detailed information if they are required. In the event that docQuery cannot answer the travelers question, the request will be sent to the expert community who will answer it.

3 The SEASALT Architecture

The SEASALT Architecture provides an application-independent architecture that features knowledge acquisition from a web-community, knowledge modularization, and agent-based knowledge maintenance. It consists of several components which will be presented in the following sections, ordered by their role within general information management, and exemplified using the docQuery project.

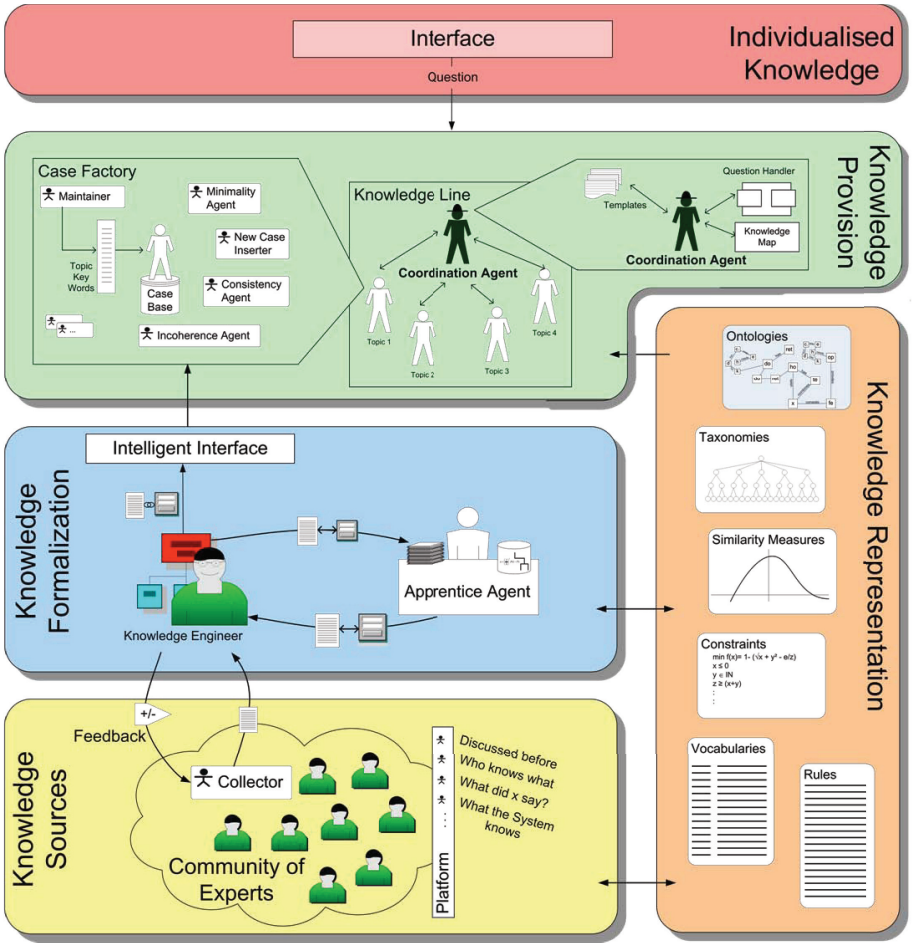

Fig. 1. The SEASALT architecture, the individual components are grouped into layers according to their function in knowledge management

3.1 Knowledge Sources

An interdisciplinary application domain such as travel medicine needs to draw information from numerous knowledge sources in order to keep up to date. Beyond traditional knowledge sources such as data bases and static web pages the main focus of SEASALT are Web 2.0 platforms. The SEASALT architecture is especially suited for the acquisition, handling and provision of experiential knowledge as it is provided by communities of practice and represented within Web 2.0 platforms [5]. Within our implementation of SEASALT we used a web forum software that was enhanced with agents for several different purposes. We chose a forum since it's a broadly established WWW communication medium and provides a low entry barrier even to only occasional WWW users. Additionally its

contents can be easily accessed using the underlying data base. The forum itself serves as a communication and collaboration platform to the travel medicine community, which consists of professionals such as scientists and physicians who specialize in travel medicine and local experts from the health sector and private persons such as frequent travelers and globetrotters. The community uses the platform for sharing experiences, asking questions and doing general networking. The forum is enhanced with agents that offer content-based services such as the identification of experts, similar discussion topics, etc. and communicate by posting relevant links directly into the respective threads such as in [6].

The community platform is monitored by a second type of agents, the so called Collector Agents. These agents are individually assigned to a specific Topic Agent (see 3.2), their task is to collect all contributions that are relevant with regard to their assigned Topic Agent's topic. The Collector Agents pass these contributions on to the Knowledge Engineer and can in return receive feedback on the delivered contribution's relevance. Our Collector Agents currently use the information extraction tool TextMarker [7] to judge the relevance of a contribution. The Knowledge Engineer reviews each Collector Agent's collected contributions and implements his or her feedback by directly adjusting the agents' rule base.

The SEASALT architecture is also able to include external knowledge sources by equipping individual Collector Agents with data base or web service protocols or HTML crawling capabilities. This allows the inclusion of additional knowledge sources such as the web pages of the Department of Foreign Affairs or the WHO.

3.2 Knowledge Formalization

In order for the collected knowledge to be easily usable within the Knowledge Line (see 3.3) the collected contributions have to be formalized from their textual representation into a more modular, structured representation. This task is mainly carried out by the Knowledge Engineer. In the docQuery project the role of the Knowledge Engineer is carried out by several human experts, who carry out the Knowledge Engineer's tasks together. The Knowledge Engineer is the link between the community and the Topic Agents. He or she receives posts from the Collectors that are relevant with regard to one of the fields, represented by the Topic Agents, and formalizes them for insertion in the Topic Agents' knowledge bases using the Intelligent Interface. In the future the Knowledge Engineer will be additionally supported by the Apprentice Agent. The Intelligent Interface serves as the Knowledge Engineer's case authoring work bench for formalizing textual knowledge into structured CBR cases. It has been developed analogous to [8] and offers a graphical user interface that presents options for searching, browsing and editing cases and a controlled vocabulary.

The Apprentice Agent is meant to support the Knowledge Engineer in formalizing relevant posts for insertion in the Topic Agents' knowledge bases. It is trained by the Knowledge Engineer with community posts and their formalizations. The apprentice agent is currently being developed using GATE [9] and RapidMiner [10]. We use a combined classification/extraction approach that first classifies the contributions with regard to the knowledge available within the

individual contributions using term-doc-matrix representations of the contributions and RapidMiner, and then attempts to extract the included entities and their exact relations using GATE. Considering docQuery's sensitive medical application domain we only use the Apprentice Agent for preprocessing. All its formalizations will have to be reviewed by the Knowledge Engineer, but we still expect a significantly reduced workload for the Knowledge engineer(s).

3.3 Knowledge Provision

SEASALT's knowledge provision is realized using the Knowledge Line approach [11]. The Knowledge Line's basic idea is a modularization of knowledge analogous to the modularization of software in the Product Line approach within software engineering [12]. Within the SEASALT architecture this knowledge modularization happens with regard to individual topics that are represented within the respective knowledge domain. Within the docQuery application domain travel medicine we identified the following topics: geography, diseases, pharmaceutics, constraints caused by chronic illnesses, vacation activities, local health facilities, and local safety precautions.

These topics are represented by Topic Agents. According to the SEASALT architecture the Topic Agents can be any kind of information system or service including CBR systems, databases, web services etc. Within docQuery we used the empolis Information Access Suite e:IAS [13], an industrial-strength CBR system, for realizing the individual agents [14]. We additionally extended the Topic Agents' CBR systems with Case Factories, which take care of the individual agents' case maintenance. The Case Factory approach is presented in more detail in [15]. Within SEASALT the Case Factory is used as a knowledge maintenance mechanism, comprising a number of agents that each carry out a simple maintenance task on an individual Topic agent's case base such as adding new cases, preserving consistency, or generalizing redundant cases [16].

The Topic Agents are orchestrated by a central Coordination Agent. The Coordination Agent receives a semi-structured natural language query from the user, analyses it using a rule-based question handler and subsequently queries the respective Topic Agents using incremental reasoning, that is using one agent's output as the next agent's input. In doing so the Coordination Agent's course of queries resembles the approach of a human amateur trying to answer a complex travel medical question. Confronted, for instance, with the question *"Which safety precaution should I take if I want to go diving in Alor for two weeks around Easter?"* and being no expert on the field an average person would first consult someone or something in order to find out that Alor is an Indonesian Island. Reading up on Indonesia the person would then find out that the rain season in Indonesia ends around Easter and that there is a hightened risk of Malaria during that time. The person would then look up information on Malaria and find out that the risk of contracting Malaria can be significantly reduced using prophylactic drugs. Knowing this he/she would the go on and acquire information on Malaria preventions and so on and so forth. This approach is mimicked by the Coordination Agent's approach. The world knowledge needed in

order to carry out this incremental reasoning process is represented within the so called Knowledge Map, which provides formal representations of all Topic Agents and possible output/input connections encoded in a graph-like structure. The Coordination Agent's implementation is described in detail in [11], its theoretic foundations are described in [17]. Finally the Coordination Agent uses the query results and prefabricated templates to compose an information leaflet to be given to the user.

3.4 Knowledge Representation

Since the miscellaneous agents operating on the community platform (see Section 3.1), the Knowledge Engineer's tools (see Section 3.2) and the CBR systems of the individual Topic Agents (see Section 3.3) deal with the same knowledge domain(s), it makes sense to join their underlying knowledge models. This does not only greatly facilitate knowledge model maintenance but also allows for an easier interoperability between the individual components. SEASALT's knowledge representation includes rules, vocabulary, ontologies, and taxonomies, some of which were handmade for the purpose of the docQuery project, some are external, such as for instance WordNet (`http://wordnet.princeton.edu/`) or ICD10 (`http://www.who.int/classifications/icd/en/`).

3.5 Individualised Knowledge

The user interacts with docQuery via a web-based interface. The web based interface offers a semi-structured input in the form of different text fields used for entering information on the destination, the traveler, the time of travel and so on. The docQuery system provides individualized knowledge to its users by generating information leaflets as PDFs that only include information which is relevant to the respective user and its journey and can be used by the traveler to consult a physician for final advice and prescriptions.

4 Evaluation

A comparative evaluation of the SEASALT architecture in general is difficult, since the tasks within the docQuery project were executed manually until we started to introduce the system. Also we think that a purely local evaluation with regard to performance and runtime would be of little value to fellow researchers. Because of this we chose to do a practical evaluation within our first application domain travel medicine. The domain required a modularization of knowledge sources because the practitioners do not only use medical knowledge, but also need for instance regional and political information. This requirement is met by the concept of the Knowledge Line, because the Topic Agents represent an expert and their collaboration the composition of information leaflets. Especially the regional and political information have to be up-to-date and therefore we are able to extract knowledge about such topics from web communities and provide

them within docQuery. The developed knowledge acquisition process optimizes the time until this information is available. Our application partner's current best practice is the manual assembling of information leaflets, mostly copy-pasting recurrent texts (like general information and warnings) from prepared templates and external sources. The application partner has been compiling these information leaflets for several years and has in the meantime optimized the process as far as possible. Using this approach a trained medical practitioner needs about an hour to create a complete leaflet. First tests have shown that the docQuery system offers a significant time saving and takes a lot of repetitive tasks from the medical practitioner. Even when counterchecking every generated leaflet and, if necessary, adding corrections or additional information the process of composition of information leaflets is significantly accelerated using docQuery.

5 Summary and Outlook

In this paper we presented the SEASALT architecture and described and exemplified its individual components using SEASALT's first instantiation, the docQuery project. The SEASALT architecture offers several features, namely knowledge acquisition from web 2.0 communities, modularized knowledge storage and processing and agent-based knowledge maintenance. SEASALT's first application within the docQuery project yielded very satisfactory results, however in order to further develop the architecture we are planning to improve it in several areas. One of these are the Collector Agents working on the community platform, which we plan to advance from a rule-based approach to a classification method that is able to learn from feedback, such as for instance CBR, so more workload is taken off the Knowledge Engineer. Also to this end more work will go into the Apprentice Agent, which is currently being developed. Another area of research that we currently look into are trust and provenance of information. SEASALT incorporates information from a large number of sources and we are currently looking into methods for making the source of the individual pieces of information more transparent to docQuery's users and thus improve the system's acceptance and trustworthiness. Finally we are planning to also apply the architecture in other application scenarios in order to further develop it and also ensure its general applicability in different application scenarios.

References

1. Althoff, K.D., Bach, K., Deutsch, J.O., Hanft, A., Mänz, J., Müller, T., Newo, R., Reichle, M., Schaaf, M., Weis, K.H.: Collaborative multi-expert-systems – realizing knowlegde-product-lines with case factories and distributed learning systems. In: Baumeister, J., Seipel, D. (eds.) KESE @ KI 2007, Osnabrück (September 2007)
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 1(7) (March 1994)

3. Bergmann, R., Althoff, K.D., Breen, S., Göker, M.H., Manago, M., Traphöner, R., Wess, S.: Selected Applications of the Structural Case-Based Reasoning Approach. In: Bergmann, R., Althoff, K.-D., Breen, S., Göker, M.H., Manago, M., Traphöner, R., Wess, S. (eds.) Developing Industrial Case-Based Reasoning Applications, 2nd edn. LNCS (LNAI), vol. 1612, pp. 35–70. Springer, Heidelberg (2003)
4. Bergmann, R., Althoff, K.D., Minor, M., Reichle, M., Bach, K.: Case-based reasoning - introduction and recent developments. Künstliche Intelligenz: Special Issue on Case-Based Reasoning 23(1), 5–11 (2009)
5. Plaza, E.: Semantics and experience in the future web. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 44–58. Springer, Heidelberg (2008)
6. Feng, D., Shaw, E., Kim, J., Hovy, E.: An intelligent discussion-bot for answering student queries in threaded discussions. In: IUI 2006: Proc. of the 11th Intl. Conference on Intelligent user interfaces, pp. 171–177. ACM Press, New York (2006)
7. Klügl, P., Atzmüller, M., Puppe, F.: Test-driven development of complex information extraction systems using textmarker. In: Nalepa, G.J., Baumeister, J. (eds.) KESE. CEUR Workshop Proceedings, CEUR-WS.org, vol. 425 (2008)
8. Bach, K.: Domänenmodellierung im textuellen fallbasierten schließen. Master's thesis, Institute of Computer Science, University of Hildesheim (2007)
9. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proc. of the 40th Meeting of the Association for Computational Linguistics (2002)
10. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Ungar, L., Craven, M., Gunopulos, D., Eliassi-Rad, T. (eds.) KDD 2006: Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, August 2006, pp. 935–940. ACM, New York (2006)
11. Bach, K., Reichle, M., Reichle-Schmehl, A., Althoff, K.D.: Implementing a coordination agent for modularised case bases. In: Petridis, M., Wiratunga, N. (eds.) Proc. of 13th UKCBR @ AI 2008, December 2008, pp. 1–12 (2008)
12. van der Linden, F., Schmid, K., Rommes, E.: Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering. Springer, Heidelberg (2007)
13. empolis GmbH: Technical white paper e:information access suite. Technical report, empolis GmbH (September 2005)
14. Althoff, K.D., Reichle, M., Bach, K.: Realizing modularized knowledge models for heterogeneous application domains. In: Perner, P. (ed.) ICDM 2008. LNCS (LNAI), vol. 5077, pp. 114–128. Springer, Heidelberg (2008)
15. Althoff, K.D., Hanft, A., Schaaf, M.: Case factory – maintaining experience to learn. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 429–442. Springer, Heidelberg (2006)
16. Bach, K.: docquery - a medical information system for travellers. Internal project report (September 2007)
17. Reichle, M., Bach, K., Reichle-Schmehl, A., Althoff, K.D.: Management of distributed knowledge sources for complex application domains. In: Proc. 5th Conference on Professional Knowledge Manegement - Experiences and Visions, WM 2009 (2009)