# RESUBMISSION: Management of Distributed Knowledge Sources for Complex Application Domains

**Meike Reichle  Kerstin Bach  Alexander Reichle-Schmehl  Klaus-Dieter Althoff**

University of Hildesheim, Dep. of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
{lastname}@iis.uni-hildesheim.de

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. This paper's focus[1] lies on the distributed knowledge sources of the SEASALT architecture [Reichle *et al.*, 2009a] and their management and (optimised) querying using a Coordination Agent [Bach *et al.*, 2008]. Within SEASALT knowledge modularisation is realised in the *Knowledge Line* that is based on the principle of product lines as it is known from software engineering. We apply this 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.

A so called Knowledge Map organises all available knowledge sources that can be accessed by a Coordination Agent that creates individual requests and combines information. The term Knowledge Map originates in Davenport's and Prusak's work on Working Knowledge [Davenport and Prusak, 2000] in which they describe a Knowledge Map from the organisational point of view mapping human experts in a large organisation or company. We transfer this concept to an intelligent agent framework that coordinates different knowledge sources.

The Coordination Agent navigates through the Map and subsequently queries the individual knowledge sources and thus creating an individual path through the map[Reichle-Schmehl, 2008]. There are dependencies between knowledge sources, a dependency exists if one source's output serves as another's input and thus enforces a subsequent query. Since the dependencies between knowledge sources can take any form, the Knowledge Map is implemented as a graph where each knowledge source is represented by a node and directed edges denote the dependencies. Retrieval paths are computed based on the information a user gives in an individual query and the properties of the knowledge sources. Our current implementation provides an a-priori computation of the retrieval path using a modified Dijkstra algorithm to determine an optimal route over the graph.

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: Access Limits, Answer Speed, Economic Cost, Syntax, Format, Structure, Cardinality, Trust or Provenance
- Content properties: Content, Expiry, Up-to-dateness, Coverage, Completeness

While these properties can be easily described and modeled, 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.

Not all of the properties presented above are fully unrelated. The properties syntax, 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 speed, syntax 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 keeps them up to date.

## References

[Bach *et al.*, 2008] K. Bach, M. Reichle, A. Reichle-Schmehl, and K.-D. Althoff. Implementing a Coordination Agent for Modularised Case Bases. In M. Petridis, editor, *Proc. UKCBR-08*, December 2008.

[Davenport and Prusak, 2000] T. H. Davenport and L. Prusak. *Working Knowledge: How Organizations Manage What they Know*. Harvard Business School Press, 2000.

[Reichle *et al.*, 2009a] M. Reichle, K. Bach, and K.-D. Althoff. The SEASALT Architecture and its Realization within the docQuery Project. In B. Mertsching, editor, *Proc. KI-2009*, LNCS, pages 556 – 563. Springer, 2009.

[Reichle *et al.*, 2009b] M. Reichle, K. Bach, A. Reichle-Schmehl, and K.-D. Althoff. Management of Distributed Knowledge Sources for Complex Application Domains. In K. Hinkelmann and H. Wache, editors, *Proc. WM2009*, LNI, pages 128–138, March 2009.

[Reichle-Schmehl, 2008] A. Reichle-Schmehl. Entwurf und Implementierung eines Softwareagenten zur Koordination des dynamischen Retrievals auf verteilten, heterogenen Fallbasen. BSc Thesis, September 2008.

[1] This is a one-page abstract for the full paper and references see [Reichle *et al.*, 2009b]