Position Paper for the First AKT Workshop on Semantic Web Services

Birgitta König-Ries[1] and Michael Klein[2]

[1] Institut of Computer Science, Friedrich-Schiller-Universität Jena, D-07743 Jena, Germany, koenig@informatik.uni-jena.de

[2] Institute for Program Structures and Data Organization, Universität Karlsruhe, D-76128 Karlsruhe, Germany, kleinm@ipd.uni-karlsruhe.de

1 The DIANE Service Description

The goal of the DIANE project[1] is it to enable automatic resource sharing in dynamic environments, more precisely in ad hoc networks. The work is based on the paradigm of service oriented computing. One major prerequisite to allow for *automatic* resource sharing in such a system is the ability to automatically discover and bind services. This in turn requires semantic service descriptions and appropriate methods to match service offers and requests.

Since existing service description languages, in particular OWL-S, did not provide all that is necessary for automation, we have developed our own description language, the DIANE Service Description (DSD). In our opinion, this language prototypically realizes all the functionality that is needed from a semantic service description.

A major enhancement of DSD over OWL-S is the explicit distinction between service offers and requests. While in OWL-S (and most other proposals), service requests are formulated as descriptions of the ideal service, DSD allows for a more flexible, yet at the same time more precise description of what is needed. Instead of specifying one instance (namely the ideal one) and leaving it to the matcher to determine how close any given offer is to that request, in DSD, a requestor will specify a fuzzy set of acceptable services. This specification explicitly encodes the user's preferences so that the matcher is able to unambiguously decide how well a service offer matches the request.

A second major difference of DSD from other approaches is its pure state-orientation. Instead of modeling message flow and state change separately as OWL-S and WSMO do, DSD describes services exclusively by their state change. The message flow is encoded in these effects by the introduction of input and output variables. These variables are bound during the matching process. This approach has two advantages: First, the semantics of the service are captured more clearly, since the influence of input variables on the effect is made explicit. Second, it allows to invoke services that offer the desired functionality but use a different interface then the one envisioned by the requestor.

[1] http://www.ipd.uka.de/DIANE/en

2 Tools from the DIANE Project

Within the DIANE project, we are developing not only the language itself, but also a number of accompanying tools.

- A Microsoft VISIO template is available that allows even unexperienced users to graphically develop DSD descriptions. These are then automatically translated into a formal representation.
- Transformation tools that transform the formal representation for example in a java based one. These java classes are then used by the other components of the system.
- A matcher that takes full advantage of the features of DSD is currently being developed. A preliminary version with limited capabilities is available, the full matcher will be realized by the beginning of 2005.
- A simulation environment, DIANEmu, that allows for extensive testing of service discovery and invocation in a dynamic environment, is available, too.
- Finally, we offer an execution framework, i.e. a middleware platform.

3 Questions to be Addressed at the Workshop and Challenges for Semantic Web Services

- What are the key features each description language for semantic web services should possess? Which of these features are still lacking from current approaches?
- Do we really need powerful reasoning mechanisms for semantic web services? Can't matching be done without them?
- A unified world ontology is certainly not realistic. How can we handle a multitude of small (and possibly overlapping) ontologies?
- Will there be "the one" description language? If not: How can co-existing solutions be used in a unified way (or: Given a request expressed in OWL-S, will I be able to find a service described in WSMO?)

About the Authors

Birgitta König-Ries got her PhD from the University of Karlsruhe in 1999. After 3 years as a postdoctoral research assistant at the University of Louisiana at Lafayette and Florida International University, she returned to Germany. After a few years as head of the mobile information systems group at Karlsruhe University, she has recently joined the Computer Science Department of the Friedrich-Schiller-University in Jena. Her research is focused on resource sharing in dynamic, in particular mobile environments.

Michael Klein studied computer science at the University of Karlsruhe in Germany. Since 2001, he is working towards his PhD at the Institute for Program Structures and Data Organization at the University of Karlsruhe. His main research area is semantic service descriptions.