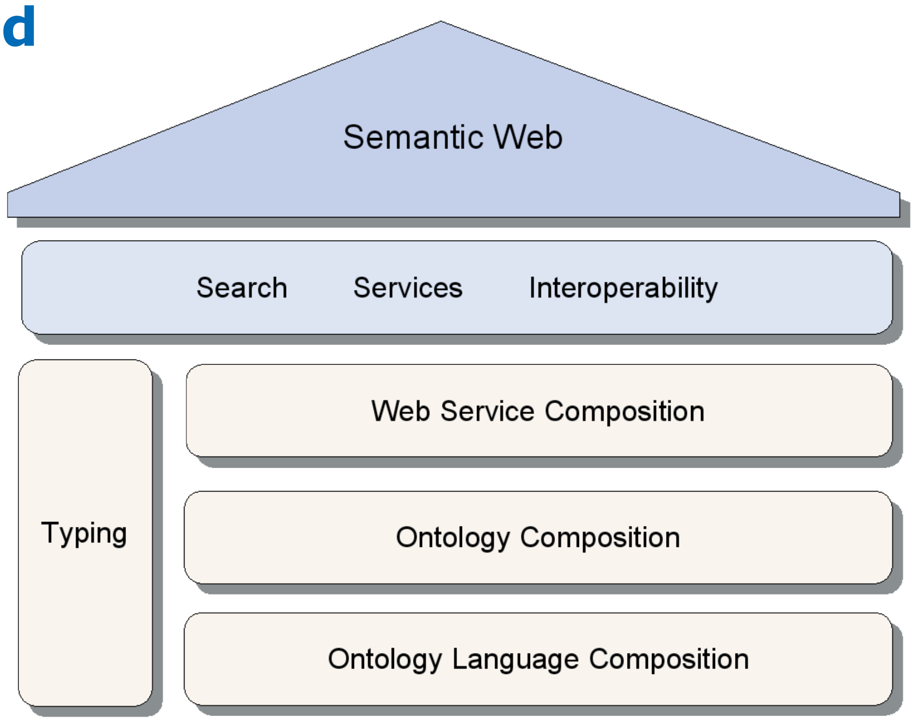
Composition and Typing



Abstract

To save costs in constructing web applications, an appropriate reuse technology must be developed. Working group "Composition and Typing" develops the foundations of a composition and typing technology for application components (services), ontology components, as well as ontology language components.

Mission

Similar to standard software products, services on the future Semantic Web will need to be composed from smaller service components and backbone frameworks. As usual, programmers are lazy. They do not want to develop their programs over and over again, but want to reuse already existing parts in new applications. They want to be more productive and quicker than the competitor. (In short, they want to earn more money than their competitors.)

Also the success of the future Semantic Web depends on that applications can be produced very quickly. To this end, an appropriate reuse technology should be developed that treats many different ontologies, and also different ontology languages. In particular, this problem is important for web services, since we would like to engineer service families instead of single services. So, how can we build product and service families for ontology-based applications?

For reuse, type systems and component models play a major role. Type systems provide reuse from the programming languages' point of view; component models provide reuse from the application point of view. Type systems facilitate automatic detection of errors, facilitate component conformance checking, support more efficient implementation, contribute to readability of programs, etc. Component models provide interface concepts for reusable units, take application concerns into consideration, and allow for composition of components to larger systems.

And this defines the mission of the working group "Composition and Typing": to investigate modern typing concepts for rule-based ontology languages, and to develop component composition technology for the Semantic Web. This will lay the foundation for a reuse technology for the Semantic Web, for product families of web applications, as well as for component-based application engineering on the Web.

Use Scenarios

One of the interesting future application fields are web shops. Although every company wants to have its own, individual, and unique shop, the costs for its construction should be low. Hence, frameworks for web shops have to be developed that can be tailored to customer requirements in a much more flexible way than today: the future frameworks have to adapt to existing and future component models, to many ontologies, and to several ontology languages. For such a framework of web shops, ontologies from all kinds of application domains must be combined. E.g., when selling medical books, medical ontologies have to be composed with library ontologies, billing ontologies with address ontologies, a.s.o. Even if these are written in different ontology languages, communication and interoperability must be guaranteed. And here, composition and typing technology will enable the reuse and interoperability on all abstraction levels.

More information available at

http://rewerse.net/i3

Description of Research

In the working group "Composition and Typing", we will lay the foundation to integrate two notions of the Semantic Web, closely related to type systems of programming languages: schema languages and ontologies. Document classes are specified by means of schema languages. Application domains are characterized by ontologies defining hierarchies of the relevant concepts and atomic types. Another objective is to develop a system of typing for the Web reasoning languages of interest for REWERSE; to obtain similar advantages as those of types for programming languages. The third objective of the working group is to develop the foundations of a composition framework for service, ontology, and ontology language components. We intend to exploit known composition technologies of software engineering, in particular the gray-box component technology.

Tools & Technologies

■ Component composition system COMPOST:

As a demonstrator, the working group develops a software framework in Java that can compose components of many different component models. This "composition system" will be able to define component models for arbitrary languages, introducing reuse concepts also for the ontology languages of REWERSE. It will be instantiated for OWL so that OWL components can be formed and reused very flexibly. Generic OWL, views on OWL, and aspects of OWL components will not be a problem, although they have not been defined in the language itself.

■ Typing engine for REWERSE languages:

Additionally, a typing engine for the rule languages of REWERSE will be developed. It will speed up evaluation, help to detect errors, and allow for cost savings in ontology development.

■ Calendar and Time Type System CaTTS:

Reconsidering the use scenario, even Web shops include data referring to cultural calendars (e.g. Gregorian, Hebrew) as well as to professional calendars defining types like business days or delivery times. Furthermore, calendar constraint reasoning with such data is necessary, e.g. for formulating and solving constraints on delivery times. The Calendar and Time Type System CaTTS, developed in I3, provides a means to model and reason with such data. CaTTS goes far beyond predefined types for dates and time of the Gregorian calendar as supported with XML Schema. CaTTS first gives rise to declaratively specifying more or less complex calendars (in terms of reusable components similar to modules). CaTTS further offers a tool for static type checking of data typed by any calendar defined in CaTTS. CaTTS finally offers a language for declaratively expressing and a solver for efficiently solving temporal constraints.

2004.11-300

Contact Person

Dr. Uwe Assmann, Professor
Chair for Software Engineering
Fakultät Informatik

Institut für Software- und Multimediatechnik
Technical University of Dresden
01062 Dresden, DE

Phone: +49 351 463 384 64
Email:uwe.assmann@inf.tu-dresden.de

www.smt.inf.tu-dresden.de

Members

Uwe Assmann, Ilie Savga (Dresden); Jan Małuszynski, Kristian Sandahl, Adrian Pop, Yuxiao Zhao and Artur Wilk (Linköping); François Fages, Pierre Deransart, Sylvain Soliman, Nathalie Chabrier, Emmanuel Coquery, Sorin Craciunescu, Rémy Haemmerlé, Ludovic Langevine (Paris); Claude Kirchner, Horatiu Cirstea, Benjamin Wack, Clara Bertolissi (Nancy); Matthew Montebello, Charlie Abela (Malta); Wlodzimierz Drabent (Warsaw); François Bry, Stephanie Spranger (Munich); Michael Kifer, Guizhen Yang, Chang Zhao (State University of New York at Stony Brook)

Impressum

webXcerpt Software GmbH
REWERSE Technology Transfer
Aurbacherstr. 2, D-81541 Munich
http://rewerse.net

Contact: Andrea Kulas
ak@webxcerpt.com
Phone: +49 89 54 80 88 48

Responsible for the content:
Prof. Uwe Assmann
Chair for Software Engineering
Fakultät Informatik
Institut für Software und Multimediatechnik
Technical University of Dresden
uwe.assmann@inf.tu-dresden.de
Phone: +49 351 463 38464