Demos and Posters REWERSE Review Meeting Year 3

May 15-16, 2007

IBZ (International Center for Science and the Humanities e.V.) in Munich

http://rewerse.net/events/review-meeting-2007/

Links to all REWERSE demos: http://rewese.net/demos/

Overview

1 Demos and Posters REWERSE Review Meeting Year 3

1.1 WG I1 – Rule Markup Languages

1.1.1 Strelka - Visual Modeling of Rules

Details: http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=Strelka

Strelka is a tool for visual modeling of rules. It is implemented as a plug-in for the Fujaba Tool Suite (http://www.fujaba.de), which is an open source UML case tool.

The main functionalities of the tool include the vizual modeling of rules in top of UML class diagrams and serialization of rules in the interchange format R2ML.

This allows, using R2ML translators (http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=translators) to deploy rules in executable platforms such as F-Logic and JBoss Rules.

1.1.2 R2ML Web Service of Rule Interchange

Details: http://oxygen.informatik.tu-cottbus.de/rewerse-i1/?q=ws

The Web Service for Rule Interchange is a service built as a JBoss application. It uses the R2ML as an interchange language.

A demo version can be reached (the endpoint URL) at

http://hydrogen.informatik.tu-cottbus.de:8080/translator/Webservice

and the WSDL description file at:

http://hydrogen.informatik.tu-cottbus.de:8080/translator/Webservice?wsdl

A simple PHP client is provided for not experienced web service users (http://oxygen.informatik.tu-cottbus.de/R2ML/0.4/WS/R2MLTranslatorWebService_PHP_client_14032007.zip)

1.1.3 Poster:The R2ML Framework

1.2 WG I2 – Policy enforcement composition and conformance

1.2.1 Attempto Parsing Engine

We show the new features of the Attempto Parsing Engine and the language extensions of ACE 5.

See the APE web-interface at http://attempto.ifi.unizh.ch/ape/.

1.2.2 ACE Plug-in for Protégé

We show how Attempto Controlled English can be used as user-friendly interface language for the Protégé ontology editor. A screen-cast is available at

http://attempto.ifi.unizh.ch/site/docs/screencast_ace_in_protege.mov.

1.2.3 AceRules

AceRules now allows for three different semantics of rules expressed in ACE. A screen-cast is available at

http://attempto.ifi.unizh.ch/site/docs/screencast_acerules.mov.

1.2.4 ACE Authoring Tool

In the context of AceWiki we show a predictive ACE editor that guides users in the composition of ACE texts. Details can be found at

http://attempto.ifi.unizh.ch/site/docs/authoring_tools.html.

1.2.5 Protune-X: Complete Explanation Facility

An illustration of the newly implemented advanced queries *why*, *how-to*, and *what-if* supported by Protune-X.

This demo shows the different verbalization modalities obtained by analyzing blurred conditions, that carry information about information completeness and the possibility of applying selectively the closed world assumption.

http://rewerse.net/I2/software.html

1.3 WP I3 – Composition and Typing

1.3.1 Xcerptware: Xcerpt module system

Responsible: Jakob Henriksson - Dresden

The goal with the composition framework – **Reuseware** – developed within I3 is to allow for layering of a *light-weight composition system* (LWCS) on top of an existing language and its associated tools. A specific LWCS provides added abstractions and reuse possibilities for the underlying targeted language. Query languages for the Web have inflexible possibilities for abstractions and poorly support modular designs, where reuse of common queries and query-parts play a role. The possibility of reuse of queries and query-parts will become increasingly important as queries become more and more complex on the inherently heterogeneous (Semantic) Web.

Xcerptware is a step towards this vision. Xcerptware provides added abstractions to the XML query and transformation language Xcerpt developed within REWESE (I4). As an example, Xcerptware implements a module system for Xcerpt, allowing users to encapsulate sets of Xcerpt rules in reusable units – *modules*.

1.3.2 Ontology composition: delayed modeling decisions via composition

Responsible: Jakob Henriksson - Dresden

Ontologies should ideally be constructed from reusable components – *composed*. We aim at layering a *light-weight composition system* (LWCS) on top of existing ontology language to provide richer abstractions and allow for the possibility to define reusable ontology components. Collaboration-based design from object-orientation and role modeling is currently being targeted to be transferred to the domain of ontologies.

As a first experimentation, however, we investigate possibilities of defining ontology components where certain modeling decisions for the resulting ontology have been postponed and left open, undecided until the decision needs to be made. Ontologies can be modeled and designed in different ways, which has an effect on how the resulting ontology is expressed and understood by fellow humans, but more importantly, it also has impacts on how the resulting ontology is understood by automatic *reasoners* (e.g. RacerPro, FacT++ etc.).

Our composition environment – **Reuseware –** allows for the delay of ontology modeling decisions and can encapsulate each decision as an atomic, user-defined, *composition operator*. Executing the composition operators realizes a design decision. As such, the ontology component may be reused in different ontologies, but also the composition operators over several ontologies.

1.3.3 PreDiCtS

http://www.cs.um.edu.mt/~cabe2/research/projects/predicts/predicts.html

Responsible: Charlie Abela - Malta

The increase in adoption of the Web services technology requires tools that provide more personalisation during the Web-service retrieval process. Personalisation is important since it is expected to lead to more precise service-discovery and faster composition. **PreDiCtS** is a framework for the personalised retrieval of service-templates. We adopt the notion that similar service-composition problems can be tackled in a similar manner by reusing and adapting past composition best practices or templates.

The retrieval process in **PreDiCtS** uses a mixed- initiative technique based on *Conversational Case-Based Reasoning* (CCBR) that provides i) for a clearer identification of the user's service requirements and ii) based on these requirements, finds suitable service templates that satisfy the user's goal.

Templates are based on our own **CCBROnto**, OWL ontology, and combine context knowledge with, sets of question-answer pairs (required by the CCBR engine) and with service-composition descriptions based on *OWL-S*.

Initial experimentation with **PreDiCtS** focused on the use of a library of *similarity metrics* during the template-retrieval process. The present library includes also a *taxonomic similarity metric* that takes into consideration the taxonomic aspect of concepts defined in the templates, to minimise the length of the retrieval process.

1.4 WP I4 – Reasoning-Aware Querying

1.4.1 Poster: RDFLog: Taming Existence–Querying RDF with b-nodes

1.4.2 Poster: GRDDLing with Xcerpt: Learn one, get one free!

1.4.3 Poster: Breaking down Web Queries: Algrebaic Evaluation of Xcerpt and Xquery

1.5 WG I5 – Evolution and Reactivity

Towards Specifying the Evolution of Web-based Data Repositories

1.5.1 MARS: Modular Active Rules for the Semantic Web

The MARS Framework provides an environment for a modular specification of behavior in the Semantic Web by ECA rules. The framework allows to embed arbitrary languages for specifying the event, condition, and action components. In addition to a communication infrastructure that basically uses XML, it allows to give a semantic account of the services and languages. Rules can optionally be specified in RDF and OWL, and then can be dealt with as semantic items themselves. The current prototype gives a basic, mainly technical, proof-of-concept of the language and service composition.

The demonstrator is accessible via

http://www.semwebtech.org/mars/frontend/

Project homepage:

http://www.dbis.informatik.uni-goettingen.de/MARS/

1.6 WG A1 – Web-based Decision Support for Event, Temporal and Geographical Data

1.6.1 Visu-L-DSMS

The Local Data Stream Management System (L-DSMS) is a by-product of A1's research in processing locational data. It had been developed to ease the implementation of systems for processing streams of traffic data. One can specify networks of processing nodes of almost any kind in an XML-file. This specification is then automatically compiled into an executable Java program.

Visu-L-DSMS is a graphical user interface (GUI) for the L-DSMS (cf. http://rewerse.net/A1/ldsms). It's purpose is to ease the management of L-DSMS installations. The final functionality will include a graphical editor for the configuration file, control functions for the subsystems, administrative functions, and visualization of input and output on node level (within subsystems). The current prototype visualizes the configuration graph and node-level input / output.

1.6.2 MPLL Protoype

This demo introduces the prototype implementation of the Multi Paradigm Location Language (MPLL). MPLL is a functional specification and programming language specifically tailored for working with geospatial notions. Various features of the prototype will be demonstrated in a telnet session.

1.6.3 Visualization and Query Processing with the TransRoute Framework

This demo presents the TransRoute framework which is the kernel of A1's geospatial information processing systems. The main purpose of TransRoute is to represent and manipulate networks of hierarchically structured graphs. Such graphs can, for example, be used to represent heterogeneous transport networks. TransRoute is basically used as a server which can be contacted to solve various kinds of problems, in particular shortest paths problems.

In order to visualize its internal working, an ontology-based visualization component has been developed. To each type of graph element, a corresponding visualization can be defined. For example, intersections and train stations are displayed differently. The way in which this mapping is done can be defined in the ontology directly. In addition, the results of graph queries posed to the TransRoute system are visualized. These can be e.g. answers to shortest path or nearest neighbor queries.

1.6.4 The CTTN System (Computational Treatment of Temporal Notions)

We demonstrate the current state of the CTTN system for specifying and operating with temporal notions. In particular the new features for specifying complex periodic temporal notions will be shown. The new XML-interface allows for standardized exchange of such specifications between different systems.

1.7 WG A2 – Towards a Bioinformatics Semantic Web

1.7.1 GoPubMed

http://www.gopubmed.org/

GoPubMed has gone a long way. It includes now background knowledge on other domains such as diseases and thus allows to answer questions, which are also interesting to lay people:

- Is research on diseases such as polio and rubella decreasing?
- Which country publishes most about leprosy?
- Which disease is rhodpsin involved in?
- Which protein is involved in sickle cell disease?
- Who is a leading researcher in celiac disease?

Additionally, GoPubMed links now to relevant wiki pages, protein names and it is also available as web search engine.

1.7.2 EagleVista

Everything is a network: Interacting proteins, co-authoring scientists, co-occurring terminology. Usually, these networks are scale-free, meaning that there are a few hubs and many low connectivity nodes. EagleVista is a plug-in for the visualisation software Cytoscape, which identifies modules in networks and unravels the hidden structures of these networks.

1.7.3 Sambo (System for Aligning and Merging Biomedical Ontologies)

http://www.ida.liu.se/~iislab/projects/SAMBO/

Sambo integrates bio-ontologies such as the GeneOntology and Mesh. It deploys novel algorithms for concept mapping.

1.8 WP A3 – Personalised Information Systems

1.8.1 Personal Reader - Configurable Web Services

Featuring:

MyEar - The Personal Reader Music Agent

MyNews - The Personal Reader News Agent

Within the Personal Reader project we already developed Web Content Reader like the Personal Publication Reader which allows browsing publications in an embedded context. We also utilized and extended the SWAD-E Semantic Portal software to provide a Personal Semantic Portal for the REWERSE project (SemPortal). Whereas these approaches are fixed in terms of the type of data that is provided, we now introduce a more generic approach: *Configurable Web Services* and the *Personal Reader Agent*.

The Personal Reader Agent is a Web Application which enables users to select, configure and call Configurable Web Services. These Semantic Web Services need a detailed description of how they can be configured and how they are accessible. According to this description the Personal Reader Agent generates an interface that allows to adjust the Web Services. Personalization functionalities, like reuse of stored configurations of Web Services which suit to the user's interest, lead to an adaptive, personal Agent.

More information and access to demo material, the MyEar and MyNews application is available at:

http://www.personal-reader.de/agent/