Poster Presentation: Xcerpt and XChange – Logic Programming Languages for Querying and Evolution on the Web

François Bry, Paula-Lavinia Pătrânjan and Sebastian Schaffert

Institute for Informatics, University of Munich, http://www.pms.ifi.lmu.de/

Motivation

The *Semantic Web* is an endeavor aiming at enriching the existing Web with meta-data and data (and meta-data) processing so as to allow computer systems to actually *reason* with the data instead of merely *rendering* it. To this aim, it is necessary to be able to *query* and *update* data and meta-data. Existing Semantic Web query languages (like DQL[2]) are special purpose, i.e. they are designed for querying and reasoning with special representations like OWL or RDF, but are not capable of processing generic Web data. On the other hand, the language *Xcerpt* [1] [2] presented in the poster is a general purpose language that can query any kind of XML data and at the same time, being based on logic programming, provides advanced reasoning capabilities. It could thus serve to implement a wide range of different reasoning formalisms.

Likewise, the maintenance and evolution of data on the (Semantic) Web is necessary: the Web is a "living organism" whose dynamic character requires languages for specifying its evolution. This requirement regards not only updating data from Web resources, but also the propagation of changes on the Web. These issues have not received much attention so far, existing update languages (like XML-RL Update Language [3]) and reactive languages [4] developed for XML data offer the possibility to execute just simple update operations and, moreover, important features needed for propagation of updates on the Web are still missing. The language *XChange* also presented in the poster builds upon the query language Xcerpt and provides advanced, Web-specific capabilities, such as propagation of changes on the Web (*change*) and event-based communications between Web sites (*exchange*).

Xcerpt: Querying Data on the Web

Xcerpt is a declarative, rule-based query language for Web data (i.e. XML documents or semistructured databases) based on logic programming. An Xcerpt program contains at least one *goal* and some (maybe zero) *rules*. Rules and goals consist of query and construction patterns, called *terms* in analogy to other logic programming languages. Terms represent tree-like (or graph-like) structures. The children of a node may be either *ordered* (as in standard XML) or *unordered* (as is common in databases).

Data terms are used to represent XML documents and the data items of a semistructured database. They are similar to *ground* functional programming expressions and logical atoms. A *database* is a (multi-)set of data terms (e.g. the Web).

Query terms are patterns matched against Web resources represented by data terms. They are similar to the latter, but augmented by *variables* (for selecting data items), possibly with *variable restrictions* (the possible bindings are restricted to certain subterms), by *partial term specifications* (omitting subterms irrelevant

[1] This research has been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net).

[2] DAML Query Language, http://www.daml.org/dql.

to the query), and by additional query constructs like *subterm negation*, *optional subterm specification*, and *descendant*.

Construct terms serve to reassemble variables (the bindings of which are specified in query terms) so as to construct new data terms. Again, they are similar to the latter, but augmented by *variables* (acting as place holders for data selected in a query) and the *grouping construct* `all` (which serves to collect all instances that result from different variable bindings).

Construct-Query Rules (short: rules) relate a construct term to a query consisting of AND and/or OR connected query terms. Rules can be seen as "views" specifying how documents shaped in the form of the construct term can be obtained by evaluating the query against Web resources (e.g. an XML document or a database). Xcerpt rules might be *chained* like active or deductive database rules to form complex query programs.

XChange: Evolution of Data on the Web

Exchanging Events on the Web. The language XChange aims at establishing reactivity, expressed by *reaction rules*, as communication paradigm on the Web. With XChange, communication between Web sites is peer-to-peer, i.e. all parties have the same capabilities and can initiate communication, and synchronization can be expressed, so as to face the fact that communication on the Web might be unreliable and cannot be controlled by a central instance.

The processing of events is specified in XChange by means of event-raising rules, event-driven update rules, and event-driven transaction rules. *Event-raising rules* specify events that are to be constructed and raised as reaction to incoming (internal or external) events.

Propagating Changes on the Web. XChange provides the capability to specify relations between complex updates and execute the updates conformly (e.g. in booking a trip on the Web, one might wish to book an early flight *and* of course the corresponding hotel reservation). To deal with network communication problems, an explicit specification of synchronization operations on updates is needed, a (kind of) control which logic programming languages lack.

Update rules are rules specifying (possibly complex) updates. The head of an update rule contains patterns for the data to be modified augmented with update operations (i.e. insertion, deletion, replacement), called *update terms*, and the desired synchronization operations.

As sometimes complex updates need to be executed in an all-or-nothing manner (e.g., in booking a trip on the Web, a hotel reservation without a flight reservation is useless), the concept of *transactions* (one or more updates treated as one unit) is supported by XChange. Transactions may be executed on user requests or as reactions to incoming events (the latter transactions are specified using *event-driven transaction rules*).

References

1. Bry, F., Schaffert, S.: Towards a Declarative Query and Transformation Language for XML and Semistructured Data: Simulation Unification. In: Proc. Int. Conf. on Logic Programming (ICLP). LNCS 2401, Springer-Verlag (2002)
2. Bry, F., Schaffert, S.: An Entailment for Reasoning on the Web. In: Proc. of Rules and Rule Markup Languages for the Web (RuleML03), Springer-Verlag (2003)
3. Liu, M., Lu, L., Wang, G.: A Declarative XML-RL Update Language. In: Proc. Int. Conf. on Conceptual Modeling (ER 2003). LNCS 2813, Springer-Verlag (2003)
4. Papamarkos, G., Poulovassilis, A., Wood, P.T.: Event-condition-action rule languages for the semantic web. In: Workshop on Semantic Web and Databases, Berlin (2003)