Reasoning on the Web: Language Prototypes and Perspectives*

Sacha Berger, François Bry, Bernhard Lorenz,† Hans Jürgen Ohlbach,
Paula-Lavinia Pătrânjan, Sebastian Schaffert, Uta Schwertel, Stephanie Spranger

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

Abstract

Reasoning on the Web is gaining in importance because of emerging Web applications such as context-adaptive Web systems (e.g. eLearning, recommender, personalised (multi-)media, and mobile information systems), Web service retrieval and composition, and Semantic Web applications of all kinds. A central issue is combining automated reasoning methods with Web languages, especially with Web query, schema, and update languages. This article reports on prototypes of various kinds combining different forms of reasoning with Web languages. The languages presented here have been conceived for both, conventional Web and Semantic Web data, assuming that future Web applications will require to freely combine data of both kinds.

1 Introduction

Reasoning on the Web is key to applications such as context-adaptive Web systems, Web service retrieval and composition, and more generally Semantic Web applications of all kinds. Context-adaptive systems rely on determining from rendering device characteristics (such as screen size and bandwidth), user models or preferences adaption criteria used for selecting, structuring, and/or rendering textual or media contents. To this aim, declarative formalisms using rule-based specifications and inference are desirable because they are declarative. Mobile Web information systems adapt Web contents to time and location in general aiming at restricting the amount of data delivered to end users.

Today, for example the modelling and processing of geospatial data has been widely investigated. Geographic Information Systems (GIS) are in use in many fields from land survey and cadastral applications to Intelligent Transport Systems (ITS) and more, albeit mostly constricted by proprietary processes and standards. Furthermore, routing and navigation systems are rather common, not only as classic applications on a PC, but also on handheld devices, mobile phones and in-car systems. Geospatial reasoning for Semantic Web applications could operate in a similar manner and therefore make direct use of available and proven concepts, algorithms, and technologies. Why should a different approach be necessary?

The key issues are *mobility*, *user interaction*, and *data modelling*. Firstly, by nature of their environment, context adaptive mobile applications have to operate mostly on lean resources, regarding e.g. computation power and bandwidth. There is no sense in downloading large amounts of data onto a mobile device and performing heavy computation locally. Secondly, classic client-server applications are rarely interoperable (e.g. hotel reservation, flight and rental car booking systems) and would have to be specially customised for individual end-user devices (e.g. webbrowser, PDA, mobile phone). And thirdly,

*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).

†Contact details: lorenz@pms.ifi.lmu.de, http://www.pms.ifi.lmu.de/mitarbeiter/lorenz

modelling of geospatial data has been and still is very much depending on the nature of the application. Proprietary and highly specialised standards cater for a number of incompatibilities and problems whenever data must be shared by different systems.

One solution is that reasoning on location data be conducted on higher level representations (see Section 5). Whenever possible, *abstract* locations, such as addresses or named entities, should be used, while fallback on low level *numeric* representations, like coordinates, should still be possible whenever necessary. Furthermore, the languages used to *define*, *query* and *update* data should be interoperable and user interaction should be independent of devices and/or interfaces used.

A typical scenario dealing with location data in a multi-modal[1] environment is described in the next section. This scenario in turn serves to illustrate the application and key functions of each of the technologies described in the subsequent sections.

2 Motivating Scenario: Search for a Pharmacy in a City.

A person is on vacation in some foreign city on a Sunday afternoon. For some reason a certain medication is needed rather urgently, therefore a handheld device with wireless access to the Web is used to send a query similar to the following: "Where is the nearest pharmacy with medication xyz in stock?". Input of the query is done (depending on device capabilities) for example via speech recognition or keyboard input, in form of natural language. After a short while the device displays one or more choices of destinations, i.e. pharmacies with suitable opening hours which have the requested item in stock. In addition, the specificities[2] of the respective routes are displayed according to a personalised ranking system. Routes are for example ordered by travel times, cost or other factors. The user either accepts the suggested "best" route or chooses one of the alternatives by hand. If no route is deemed suitable by the user (or the routing system), user preferences or search criteria could be readjusted. While following the route, navigation instructions are given step by step as natural language, optionally aided by a graphical map, until the destination is reached.

To form a foundation for compound applications such as described above, interoperable technologies and systems are needed which cater for seamless and transparent integration. The following comment on the technical aspects of the scenario indicates how technologies currently under development aim to make such compound applications possible.

Facilitating querying by natural language, without adhering to overly complex syntax and/or vocabulary is the aim of **VOXX** (see Section 7). A (controlled) natural language query could be translated for example to a valid **Xcerpt** (see Section 3) query; similarly, results could be translated back to natural language (see examples in Section 7). Accurate and up to date data about public and private transportation systems are crucial for location reasoning, e.g. by **MPLL** (see Section 5), in order to produce valid destinations and optimal routes. Highly dynamic effects such as congestions in traffic flow and delays or changes in train schedules create complex routing problems. Providing mechanisms to propagate changes and events throughout the Web is the aim of **XChange** (see Section 4). Processing all kinds of temporal and calendric notions, such as opening hours and timetables, culminating in different time zones, calendar systems, daylight saving times, leap seconds/years, sunrise and sunset etc. becomes necessary at many different stages in different scenarios. **WebCal** (see Section 8) and **CaTTS** (see Section 9) provide services for reasoning on temporal and calendric data.

3 Xcerpt

Xcerpt [5] is a rule-based, declarative query and transformation language for XML data. In Xcerpt, queries and the (re-)structuring of answer (also called "constructions") are expressed in terms of patterns instead of path navigations (like in XSLT and XQuery). Pattern queries are evaluated against XML documents

[1] In this context, multi-modality pertains to different modes of transportation.

[2] These include for example the modes of transportation used, overall travel time, cost, etc.

using a non-standard form of unification (called “simulation unification” [4]). Furthermore, Xcerpt supports so-called rule chaining (with recursion), which makes it suitable even for complex query programs. Due to its foundations in logic programming, Xcerpt can also serve to implement reasoning algorithms for the Semantic Web. [2].

4 XChange

Many resources on the Web and the Semantic Web are dynamic in the sense that they can change their content over time. E.g. a Web-based information system on city transportation could report delays on departures or arrivals, and other changes of the timetables. A Web-based personalised organiser might be conceived so as to automatically react to such changes affecting its owner. A delayed arrival might result in changing a previously chosen route perhaps by using other means of transportation if available. Thus, a strong motivation exists for having means to specify and/or request updates to (local or remote) Web sites and to propagate updates over related Web resources.

XChange [2] is a high-level language for programming reactive behaviour on the Web. It builds upon the query language Xcerpt and provides advanced, Web-specific capabilities, such as propagation of changes on the Web (*change*) and event-based communication between Web sites (*exchange*). XChange offers *complex updates* (e.g. when making reservations on the Web, one might wish to book a flight *and* a train ticket, *or* only a night train ticket) that can be executed in an *all-or-nothing* manner as XChange has a concept of *transactions*. Moreover, such updates can be executed automatically as reaction to situations, i.e. *composite events* (e.g. an underground train does not operate on Sundays *and* a tram has a delay because of a car accident).

5 MPLL: Multi-Paradigm Location Language

Reasoning on locations normally operates on a numerical level (e.g. coordinates) or on a symbolic level (e.g. graphs). Extensive research has been conducted in either case [7], hence there is a broad choice of proven sets of calculi and algorithms to solve the respective tasks. As stated in section 1, whenever possible, reasoning should be conducted on higher level data, which is usually less heavy on the device’s resources.

The fundamental insight is that many queries pertaining to location information are closely related to the problem of route planning and way-finding, which reduces the problem domain considerably. There are two reasons for this. (1) Whenever a certain location is sought after, chances are, that the inquirer intends to visit the location. Cases like these result in classic route planning tasks. (2) When people refer to the “distance” between two locations in the sense of locomotion, nearly never are they talking about *distances* per se (metres, kilometres) but the *time* needed to overcome these distances (“a ten minute walk” or “a half hour by train”). In fact, in many scenarios the exact distance between two points has a rather subordinate meaning from a traveller’s viewpoint – especially in urban environments.

MPLL (Maple) aims for producing a set of reasoning techniques and for providing a framework for Sematic Web applications. This framework will include suitable data types and ontologies for processing and presenting location data.

6 Graph Grammars

Typing and schema languages based on tree grammars [1] are widely accepted in the XML and Web community. In particular DTD, XML Schema and Relax NG are based on regular tree grammars. The purpose of such formalisms inspired by tree grammars range from documentation, data validation to type checking of query and transformation languages such as XQuery and XSLT.

Although graph structured XML data – built using reference mechanisms such as ID/IDREF – is widespread, common schema languages have no notion of typed references. In other words, formalisms based on tree grammars do not fully convey the graph structures of XML and semistructured data. Graph grammars aim at filling this gap. Regular (rooted) graph grammars are proposed as an extension of regular tree grammars, providing means to explicitly model typed references and dereferenciation in XML and semistructured data.

Regular graph grammars are usable as schema formalism for XML and semistructured data containing references (e.g. the translation of ACE sentences in XML as mentioned earlier in the context of VOXX) and as datatype formalism for typed variants of Xcerpt and XChange.

7 VOXX: Verbalisation of XML and Xcerpt

As devices become smaller spoken natural language becomes increasingly attractive to interact with mobile Web applications. Moreover, unlike formal languages like XML natural language is familiar to many end-users which increases its acceptance. However, full computational processing of complex natural language sentences is in most contexts not yet realistic.

The project VOXX therefore uses a *controlled* natural language to express Web queries and Web data. This is called *verbalisation* of Web queries and data. A controlled natural language[3] is a subset of a natural language whose grammar and vocabulary has been restricted in order to reduce and/or eliminate both ambiguity and complexity. VOXX works with (adaptations of) the controlled natural language Attempto Controlled English (ACE)[4] developed at the University of Zurich. ACE can be automatically and unambiguously translated into first-order logic. Since Web data and queries are based on different formal languages (like XML or Xcerpt) VOXX builds a formal bridge by defining a common underlying formal language based on logical relations of different arities to and from which both XML or Xcerpt constructs and ACE sentences can be unambiguously mapped. Thus VOXX can work with ACE as a user-friendly interface language for the following applications: (1) Express XML data directly in the controlled language ACE (*The pharmacies A and B are in Munich. A's address is Hauptstrasse 34. The pharmacy A has the medication xyz in stock.*); (2) Use ACE as a verbalisation of the Web query language Xcerpt and thus query the Web using ACE (*Where is a pharmacy in Munich that has the medication xyz in stock*); (3) Translate existing XML documents into ACE, for example to answer queries (*Yes. Pharmacy A has the medication xyz in stock*).

8 WebCal

WebCal [8, 9, 3] is a computer program and Web server which provides advanced calendric calculations for Web services. It can represent and manipulate time points, crisp and fuzzy temporal intervals, labelled time partitionings for representing periodic temporal notions, temporal durations, calendar systems with all their peculiarities, and a lot more. With the geotemporal specification language GeTS it has a powerful facility for specifying complex parameterised temporal notions.

9 CaTTS

CaTTS [6] is a static typed calendar and time language that allows for declarative modelling of various calendars (e.g. Gregorian calendar, business calendars, holiday calendars, etc.) as types and time tabling problems over calendar and time data defined by some CaTTS calendar using its object constraint language. CaTTS is designed to extend any programming language or (Semantic) Web query language in terms of a library integrated into its host language.

CaTTS is powerful and expressive enough to define most calendars in use today. The object constraint language of CaTTS enables declarative modelling of time tabling problems which will be solved by means of temporal constraint reasoning.

10 Perspectives

This article has addressed the issue of "reasoning on the Web" from the angle of a practical scenario and sketched a couple of research projects contributing to this issue and to realising applications such as that

[3] `http://www.ics.mq.edu.au/~rolfs/controlled-natural-languages/`

[4] `http://www.ifi.unizh.ch/attempto/`

of the given scenario.

This scenario demonstrates that unconventional reasoning capabilities are likely to become increasingly important for Web-based mobile applications. In particular, forms of location reasoning combining computing and/or reasoning with coordinates and symbolic reasoning on more abstract notions such as “neighbourhood” are needed for such applications. Specifying and reasoning with temporal, and more precisely calendric data, is another issue that is likely to gain in importance for Web-based mobile applications. The calendar type language CaTTS, the WebCal system and the described approach towards location reasoning for Web applications are building blocks towards systems easing the implementation of location, temporal and calendric reasoning systems for the Web.

Because of the importance of reasoning on the Web, query and reactivity languages capable of inference are further issues gaining in importance for Web-based systems. The query and reactivity languages Xcerpt and XChange combine advanced querying or reactivity facilities with reasoning capabilities. Graph Grammars is an approach towards enhancing query and/or reactivity languages with advanced data modelling (or schema) facilities well-suited for Web applications.

Finally, VOXX is an approach towards tractable and reliable natural languages interfaces to the Web using a so-called ‘controlled English language’. Arguably, such interfaces to the Web are likely to become widespread with cellular phones and mobile computers of all kinds.

References

[1] A. Brüggemann-Klein. Formal Models in Document Processing. Technical report, Faculty of Mathematics at the University of Freiburg, 1993.

[2] F. Bry, T. Furche, P.-L. Pătrânjan, and S. Schaffert. Data Retrieval and Evolution on the (Semantic) Web: A Deductive Approach. In *Workshop on Principles and Practice of Semantic Web Reasoning*. Springer, 2004.

[3] F. Bry, B. Lorenz, H. J. Ohlbach, and S. Spranger. On reasoning on time and location on the web. In N. H. F. Bry and J. Malusyński, editors, *Principles and Practice of Semantic Web Reasoning*, LNCS 2901, pages 69–83. Springer, 2003.

[4] F. Bry and S. Schaffert. 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, 2002.

[5] F. Bry and S. Schaffert. Querying the Web Reconsidered: A Practical Introduction to Xcerpt. In *Int. Conf. Extreme Markup Languages*, 2004.

[6] F. Bry and S. Spranger. Towards a Multi-Calendar Temporal Type System for (Semantic) Web Query Languages. In *Workshop on Principles and Practice of Semantic Web Reasoning*. Springer, 2004.

[7] M. T. Escrig, F. Toledo, and A. P. del Pobil. An Overview to Qualitative Spatial Reasoning. In N. Piera, editor, *Current Trends in Qualitative Reasoning and Applications*, pages 43–60. International Center for Numerical Methods in Engineering, Barcelona, 1995.

[8] H. J. Ohlbach. Relations between fuzzy time intervals. In *Proc. of the 11th International Symposium on Temporal Representation and Reasoning*, pages 44–51. IEEE, 2004.

[9] H. J. Ohlbach. The role of labelled partitionings for modelling periodic temporal notions. In *Proc. of the 11th International Symposium on Temporal Representation and Reasoning*, pages 60–63. IEEE, 2004.