

Rule Markup Techniques for the Semantic Web

Rules have traditionally been used in theoretical computer science, compiler technology, databases, logic programming, and AI. The Semantic Web is a new W3C Activity trying to represent information in the World Wide Web such that it can be used by machines not just for display purposes, but for automation, integration, and reuse across applications. Rule markup in the Web has become a hot topic since rules were identified as a design issue of the Semantic Web. However, rule markup for the Semantic Web has not been studied as systematically as the corresponding ontology markup. This Dagstuhl Seminar was an attempt to fill the gap by bringing together researchers exploring rule systems suitable for the Web, their (XML and RDF) syntax, semantics, tractability/efficiency, and transformation/compilation. Both *derivation rules* (also called "inference rules") and state-changing *reaction rules* (also called "active" or "event-condition-action" rules), as well as any combinations, have been of interest to this effort.

This seminar has succeeded in bringing together leading researchers from the classical logic programming and knowledge representation community and from the Semantic Web community. The discussions at the seminar have been very productive, both scientifically and in terms of triggering new research activities such as a EU FP6 Network of Excellence initiative.

H. Boley (DFKI Kaiserslautern, D)
B. Grosz (MIT Cambridge, USA)
S. Tabet (Nisus Inc., USA)
G. Wagner (Eindhoven Univ. of Technology, NL)

Table of contents

Pitfalls and Practicalities of Reasoning on the Web	3
Grouping Constructs for Semistructured Data	3
Efficient Defeasible Reasoning	3
The Web Service Modeling Framework WSMF	4
Ontology Reconciliation	5
Description Logic: Axioms and Rules	5
The Design of j-DREW, a Deductive Reasoning Engine for the Semantic Web	6
GOL: A General Ontological Language	6
Logic, Approximation and Truth	7
Semantic Patterns	8
Decomposing Ontology into an Ontology Base and its Interpretation Layer	8
Integrating RuleML and OO	9
Experiences Using Rules with DAML+OIL [1]	9
On the Role of Rule Markup Techniques in ontology-based Knowledge Management	10
Mapping technology for business integration	11
Arguments and Misunderstandings: A Fuzzy Approach to Conflict Resolution in Open Systems	12
Security Mark-up and Rules	12
Deductive Databases and XML	13
Rules for Efficient XPath Evaluation	13
Semantically encoding mathematics	14
XML Data Integration using XPathLog and LoPiX	15
ER Modeling and XML Modeling	16
W4 Well-founded Semantics for the World Wide Web	16
A Rule-based Language for Querying and Transforming XML Data	17

Pitfalls and Practicalities of Reasoning on the Web

Eric Prud'hommeaux

eric@w3.org

MIT

Laboratory for Computer Science

Good semantic web design entails following design principles of the conventional web. These principals shape a growing number of semantic web tools integral to W3C's infrastructure. Experience gained from developing these tools has revealed several expected advantages, however, some inconvenient artefacts of web architecture have constrained the design. Creation and use of semantic web data in general requires that one be mindful of these issues.

Grouping Constructs for Semistructured Data

François Bry

bry@pms.informatik.uni-muenchen.de

Ludwig-Maximilians-Universität München

Institut für Informatik

Markup languages for semistructured data like XML are of growing importance as means for data exchange and storage. In this paper we propose an enhancement for the semistructured data model that allows to express more semantics and to enhance query answering. A data model is proposed and the implications on pattern matching are investigated.

Cf. <http://www.pms.informatik.uni-muenchen.de/publikationen/#PMS-FB-2001-7>

Efficient Defeasible Reasoning

Grigoris Antoniou

ga@tzi.de

Universität Bremen

FB 3 - Mathematik/Informatik

Rule-based systems have been studied for many years now, and they have found significant application in practice. But there exist other kinds of rule systems that are nonmonotonic. Such systems are important in practice because they can deal with inconsistencies in a declarative way, and because they model naturally phenomena like exceptions and priorities. Nonmonotonic systems are generally useful in cases where the available information may be incomplete.

Their basic idea is to use rules which may be defeated by other rules. In case two rules with contrary heads are potentially applicable, neither will actually be applied, unless the conflict can be resolved. A conflict among rules can be resolved using priorities on rules.

In the past few years, such systems have attracted significant attention in the nonmonotonic reasoning community, e.g. courteous and defeasible logics. Their use in various application domains has been advocated, including the modelling of regulations and business rules, modelling of contracts, legal reasoning and agent negotiations. In fact, defeasible reasoning (in the form of courteous logic programs) provides a foundation for IBM's Business Rules Markup Language and for planned W3C activities on rules.

In our talk we discuss the basic ideas of defeasible reasoning and some of its key properties. One important advantage of such systems is their focus on implementability and their low computational complexity; in the basic case, the complexity of defeasible logic is linear.

On the semantic web, our vision is that nonmonotonic rule systems will be used in applications where the available information may be incomplete. We discuss some scenarios to make this idea clear. Also we outline how nonmonotonic rule systems can be defined "on top" of an ontology layer; that is, the antecedents of rules may include predicates defined in an ontology.

The Web Service Modeling Framework WSMF

Dieter Fensel

dieter@cs.vu.nl

Vrije Universiteit Amsterdam

Dept. Math. and CS

Web Services will transform the web from a collection of information into a distributed device of computation. In order to employ their full potential, appropriate description means for web services need to be developed. For this purpose we define a full-fledged Web Service Modeling Framework (WSMF) that provides the appropriate conceptual model for developing and describing web services and their composition (complex web services). Spoken in a nutshell its philosophy is based on the following principle: maximal de-coupling complemented by scalable mediation service.

Ontology Reconciliation

Leon Sterling

leon@cs.mu.oz.au

The University of Melbourne

Dept. of Computer Science & Software Engineering

This talk builds on the premise that it is important to preserve cultural diversity in software agents and it will be impossible to agree in general on a standard ontology for agents. I argue that there is a need for 'lightweight', purposive ontologies that are made explicit for individual agents. These are represented as logic programs. I discuss some initial research on how agents with different ontologies can reconcile their differences, giving examples from classified ads and sports scores.

Description Logic: Axioms and Rules

Ian Horrocks

horrocks@cs.man.ac.uk

The University of Manchester

Dept. of Computer Science

Ontologies are set to play a key role in the ``Semantic Web'', extending syntactic interoperability to semantic interoperability by providing a source of shared and precisely defined terms. DAML+OIL is an ontology language specifically designed for use on the Web; it exploits existing Web standards (XML and RDF), adding the familiar ontological primitives of object oriented and frame based systems, and the formal rigor of a very expressive description logic. The logical basis of the language means that reasoning services can be provided, both to support ontology design and to make DAML+OIL described Web resources more accessible to automated processes. DAML+OIL ontologies consist of a set of axioms asserting, e.g., subsumption relations between classes. Rules can be seen as just another form of axiom, and vice versa. Description logic style axioms, however, have different capabilities and restrictions from those derived from horn clause rules. Unfortunately it turns out that the intersection of the two formalisms is a rather strange and asymmetrical language.

The Design of j-DREW, a Deductive Reasoning Engine for the Semantic Web

Bruce E. Spencer
Bruce.Spencer@nrc.ca
University of New Brunswick
Faculty of Computer Science

Our original idea was to provide a tool so that a computer science student with only a basic knowledge of building trees in Java and an understanding of backtracking search could build a Prolog-like engine. So we created a Java application programmer's interface to handle Prolog clause selection, binding of variables and unbinding variables when backtracking, all "behind the scenes". For efficiency, we incorporated powerful techniques from competitive theorem provers, such as discrimination trees and flatterms. Then we found that a large number of prototypes could be created quickly from these basic tools, including a search-complete Prolog with iterative deepening, Prolog with sound negation, systems for abduction and default reasoning, a RuleML engine, and a forward reasoning definite clause systems. In all cases the top-level control is a Java program, readable to an undergraduate. Future prototypes include client-server systems incorporating j-DREW, and systems that annotate conclusions with digital signatures.

The Semantic Web calls for reasoning systems using at least classical, rule-based logic, i.e. clausal first-order logic, but the call is not precise about whether systems are forward or backward reasoning, whether they need default logic, whether higher order logic can play a role. So as the Semantic web is being defined, reasoning systems need to be designed to be flexible, to cover a large class of reasoning problems, and to have access to many Web-based libraries. We think the j-DREW design has the potential to meet these needs.

GOL: A General Ontological Language

Heinrich Herre
herre@informatik.uni-leipzig.de
Universität Leipzig
Institut für Informatik

Every domain-specific ontology must use as a framework some upper-level ontology which describes the most general, domain-independent categories of reality. In the presentation we sketch a new type of upper-level ontology, which is intended to be the basis of a ontology building language GOL (for: 'General Ontological Language'). It turns out that the upper-level ontology underlying standard modelling languages such as KIF, F-Logic and CycL is restricted to the ontology of sets. Set theory has considerable mathematical power and great flexibility as a framework for modelling different sorts of structures. At the same time it has the disadvantage that sets are abstract entities (entities existing outside the realm of time, space and causality), and thus a set-theoretical framework should be supplemented by some other machinery if it is to support applications in the ripe, messy world of concrete objects. We partition the entities of the real world into sets and urelements, and then we introduce several new ontological relations between these

urelements. In contrast to standard modelling and representation formalisms, the concepts of GOL provide a machinery for representing and analysing such ontologically basic relations. GOL is thus a genuine extension of KIF and of similar languages.

Logic, Approximation and Truth

Michael M. Richter
richter@informatik.uni-kl.de
Universität Kaiserslautern
FB Informatik, AG Künstl. Intelligenz

Knowledge based systems traditionally have been logic oriented: Some abstraction of the real world took place and was formulated in some representation language; an inference mechanism was drawing the consequences. In the recent years it was the tendency to give up completeness and correctness of the knowledge base; it was also accepted that knowledge was up to changes. In applications in the Web, e.g. in E-commerce the additional difficulty is that even the goal of some search is not clearly stated; e.g. one is looking for a product from which only some property is known.

The technique which is proposed here is similarity based reasoning, developed in CBR (Case-Based Reasoning). An overview over the changes of the role of similarity in CBR in the last decade is given. It starts with the naive period where past experiences were used for actual problems on the base similar looking problem situations. The last, generalized, period is characterized that arbitrary objects are compared by similarity measures. The semantics connected with the measure is always reduced to the notion of utility. E.g. a product is similar to a functionality if the product can perform the intended functionality well. This aspect is central in e-commerce. What takes place here is semantic unification (e.g. between the product and the functionality); what is new is that we deal only with an approximation of semantic unification.

This is considered as an example for situations where pure logical solutions are not available: Approximation extends logic. Approximation has here three major aspects: To live with errors, to control the errors, to improve the solution, i.e. to reduce the error. It is pointed out how techniques from CBR can be employed here.

Semantic Patterns

Steffen Staab
staab@aifb.uni-karlsruhe.de
Universität Karlsruhe
Institut AIFB

Interoperability is one of the major design objectives when building applications for B2B and Semantic Web applications. In this talk, we have presented a methodology for engineering semantic knowledge such that these semantic structures are easier reusable when switching between several representation languages. For this purpose, we reconsider the commonalities of representation languages and their usage in actual applications. Out of this consideration we derive semantic patterns as a means to communicate knowledge at an epistemological level of representation and as a means for (partial) execution by any particular implementation of any representation language. The underlying method we propose combines the advantages of formal specification methods (where feasible) with informal, natural language explanations such as used in software engineering for design patterns.

Decomposing Ontology into an Ontology Base and its Interpretation Layer

Mustafa Jarrar
mjarrar@vub.ac.be
Vrije Universiteit Brussel
DINF (Computer Science)

We claim to achieve methodological and implementation advantages by moving the rules out from an ontology into a separate "layer", which we call Interpretation Layer. We define ontologies in a logical sense, i.e. as representationless mathematical objects that constitute the range of interpretation mappings from a first order language (called here a conceptual schema) assumed to lexically represent an application, to a set of possible ("plausible") conceptualizations of the real world domain. Following common model-theoretic database practice we propose to decompose such resources into ontology bases and instances of their explicit interpretations; the latter are seen as forming a separate layer mediating between the ontology base and the application instances committing to the ontology. In this way important methodological and other practical benefits can be achieved, such as:

- Increased independence of the construction of ontology bases from their applications;
- Acceleration and simplification of the ontology building process, since domain rules and constraints which are application-dependent, complex and therefore difficult to agree on are moved outside the ontology base;
- Better management of the communication between the applications and the ontology, as the interpretation layer can be organized into set of interpretational views, where each application can create (reuse/inherit) its own interpretational view(s);

- The use of different rule languages coexisting around a same ontology, e.g. expressed alternatively as business rules, Java classes, ... or even a mix of these;
- The availability of powerful database systems for the management of large scale ontology bases and transactions on them.

Keywords: Ontology, Semantic Web, interpretation layer, Business rules, ORM

Integrating RuleML and OO

Jens Dietrich

jens.dietrich@unforgettable.com

Polytechnic of Namibia

Information Technology

RuleML is a format that should facilitate the exchange and the persistency of rules. Considering the current (enterprise) computing landscape, most of the systems interfacing with RuleML data are either object-oriented applications (like EJB middleware) or (relational) databases. We investigate a couple of conditions a rule interchange format should meet in order to support smooth integration with those systems. In particular we discuss the following issues:

- Facts are not represented as single facts but as (descriptive) sets of clauses
- "On the fly" integration of data sources
- Typing and the use of the semantics of interfacing systems
- Declarative exception handling in rule bases.

The Mandarax open-source package (www.mandarax.org) is used to demonstrate some of the features discussed.

Experiences Using Rules with DAML+OIL [1]

Mike Dean

mdean@bbn.com

BBN Technologies

DAML+OIL [2] is a key language for defining ontologies for the Semantic Web, but does not presently support the definition of rules. We have successfully used DAML+OIL with RuleML 0.7 and native rule representations for several small applications. We will report on this experience, draw some examples from the DAML+OIL representation of the seminar agenda [3] presented yesterday, and conclude with some comments on rules and DAML+OIL.

We will also use the Semantic Web to display the home cities for (most of) the seminar participants on a world map, and present a vision for the Semantic Web.

[1] <http://www.daml.org/2002/02/dagstuhl-rules/Overview.html>

[2] <http://www.daml.org/2001/03/daml+oil-index>

[3] <http://www.daml.org/2002/02/dagstuhl-demo/dagstuhl-02061>

On the Role of Rule Markup Techniques in ontology-based Knowledge Management

Nenad Stojanovic
nst@aifb.uni-karlsruhe.de
 Universität Karlsruhe
 Inst. für Angewandte Informatik

In clinical therapy a lot of knowledge is product of experiences (past cases) and is summarized in clinical studies in the form of tables, graphics or well-structured abstracts. Consulting such knowledge is very useful support for decision making in inexperienced medical doctors, or for medical experts in some rare, ambiguous cases. Also the vendors of used medicines can benefits from searching such knowledge. The situation is the same in the dentistry or pharmacology.

Sharing such knowledge in an efficient way is a rare difficult problem: (i) the existing search-resources, as MEDLINE or ACP journal club use some controlled vocabulary (for example Medical Subject Headings – MeSH, in the case of MEDLINE), but they are truly keyword-based and do not allow complex logical statements as “treatment of epilepsy in old male patient...” (ii) a lot of clinical studies is represent in the form of tables, which are not interpretable for keyword searching.

In this paper we give a solution for such knowledge management problems by describing that knowledge in clausal/conditional, If-Then, form. Those statements are then represented in the RDFS format, so that they can be processed by machine agents. The used vocabulary is defined in the form of a medical ontology in order to enable search based on semantic means. For example, part of the content of table 1, can be represented in our framework in the following way (at the logical level): $\forall x \text{ patient}(x) \wedge \text{male}(x) \wedge \text{ages}(x, \text{middle}) \wedge \text{disease}(x, A) \wedge \text{therapy}(x, \text{surgery}) \rightarrow \text{side-effect}(x, B)$

Table1: Treatment of disease A using various therapies: experimental results:

male/female	age	therapy	prognosis	side-effect
M	middle	surgery	70%	B
F	middle	surgery	95%	B
F	old	surgery	95%	C

It is clear that capturing, representation and search for such statements in a knowledge management scenario require well-defined management process, supported by corresponding tools. In this paper we describe an application architecture for maintaining (creating, capturing, storing, searching) such rules represented in the RDFS format as well as for using them in web documents content description (semantic annotation) in order to enable more efficient search for relevant documents. The architecture is based on our existing Semantic Web technologies (KAON, SOEP, OntOMat,

Ontobroker), which are slightly extended in order to incorporate more expressible data format (rules).

The benefits of the proposed approach are manifold: integration platform for various rules sources and rules format, describing content of web resources using more expressive, clausal, format, the possibility to build high-specified rule bases by searching for relevant rules in various repositories, context-sensitive searching, to name but a few.

Our small case study shows that the clinical knowledge, represented in the form of tables and graphics, can be reused in our approach more efficiently and that the efforts to produce such representation are acceptable.

Mapping technology for business integration

Borys Omelayenko

borys@cs.vu.nl

Vrije Universiteit Amsterdam

Dept. of Mathematics & Computer Science

One of the objectives of building the future Web is to allow different enterprises to easily communicate and exchange procurement information. Existent technologies for the business-to-business area are primary targeted at large companies, and are too expensive for small and medium enterprises. These are currently starting to use XML-based document standards to represent their documents and need an efficient transformation and integration technology. The standard XML transformation language XSLT has appeared to be not sufficient for their transformation needs, and we propose a novel architecture on top of XSLT that should match their integration needs. The architecture separates four integration tasks: process, message, document, and vocabulary alignment, performed via four mediating ontologies. We propose a mapping ontology to be used to perform the alignments, and present an application scenario for doing them. This research brings a number of open questions and research directions to be discussed at the workshop.

Arguments and Misunderstandings: A Fuzzy Approach to Conflict Resolution in Open Systems

Michael Schroeder
msch@soi.city.ac.uk
City University
Dept. of Computing
School of Informatics

Expressive knowledge representation will be an important feature of the semantic web. Facts as represented in a table of a relational database or by RDF can be extended along three axes:

1. Rules & deduction can be added: the very core of RuleML.
2. Negation can be added to express positive and negative knowledge
3. Fuzziness can be added to express uncertainty.

How can we reason with these extensions? We will use argumentation as an elegant mechanism to define the semantics of rules, negation, and fuzziness. Arguments attack each other and an argument is acceptable if it can be defended against any attack. Depending on which kinds of attack and defence we allow a different notion of justified arguments results. We will present a framework to relate these notions and show which ones are identical and which ones relate to other approaches such as put forward by Dung, Prakken and Sartor, and WFSX by Alferes, Damasio, and Pereira. Next we show how to interpret fuzziness in the light of negation and define an appropriate semantics.

With such expressive knowledge representation at hand, can we apply and use it on the semantic web? The semantic web is open and the use of different ontologies will lead to misunderstandings, since concept names and predicate arity etc. may mismatch. To this end, we use our approach and embed fuzzy unification into it. Fuzzy unification unifies any atoms, but scores their similarity. This enables us to reason in the light of missing and mismatching terms and predicate names.

Security Mark-up and Rules

Grit Denker
denker@csl.sri.com
SRI International
Computer Science Laboratory

Security will play a vital role in the success of the Semantic Web. It is essential that we have tools and techniques in place that will enable us to store, manipulate, and process the information on the Semantic Web in ways that meet security requirements such as authentication, authorization, and data integrity, among others. We have proposed a core security ontology that enables us to mark-up access control restrictions and data integrity of web pages. Our ontology taps into the XML signature standard.

Security mark-up is not meaningful by itself. It is motivated by web applications that implement the various security techniques to protect data that is exchanged in transactions. A user can make the decision as to whether the application or web service meets his or her security requirements based on the security mark-up. Similarly, software agents that are equipped with a user policy can select web services on the basis of their security annotation. The DAML services ontology (DAML-S) provides means to mark-up web services. We specialized DAML-S services to express authorization processes and data integrity checks.

In the future we will extend our work to express basic user security policies as well as basic trust and delegation logics.

Deductive Databases and XML

Dietmar Seipel
seipel@informatik.uni-wuerzburg.de
Universität Würzburg
Institut für Informatik

The Semantic Web wants to provide a language for representing and reasoning with rules from various existing knowledge-representation systems.

In this talk, we first discuss about some alternative ways of representing deductive database rules - which may contain function symbols, negation and disjunction - in XML. This contributes to the RuleML initiative, which is working on standards for representing rules - e.g. business rules or logical deduction rules - in XML.

Secondly, we propose a language for handling XML data – or semi-structured, complex objects in general - in logic programming and deductive databases. This representation of complex objects is based on the concept of association lists. The subcomponents of a complex object can be accessed by path expressions, which can include aggregation operators. It is also possible to modify arbitrary subcomponents of an object by simple assignment statements using path expressions. The handling of complex objects is implemented as a library in Prolog, which can be embedded into any Prolog application. We handle XML elements by representing their attributes and their subelements as two separate association lists.

This gives us a logical language for dealing with XML data, which has querying capabilities known from XPath and XML Query, and transformation features similar to XSLT.

Rules for Efficient XPath Evaluation

Dan-Alexandru Olteanu
olteanu@informatik.uni-muenchen.de

Ludwig-Maximilians-Universität München
Institut für Informatik

The location path language XPath is of particular importance for XML applications since it is a core component of many XML processing standards such as XSLT or XQuery. This work shows how, based on axis symmetry of XPath, equivalences of XPath 1.0 location paths involving reverse axes, such as ancestor or preceding, are established. These equivalences are used as rewriting rules in an algorithm for transforming location paths with reverse axes into equivalent reverse-axis-free ones. The rules are classified into two rule sets, a general one and a specific one. The general rule set treats the removal of reverse axes in the same way, regardless of the specificity of each axis. Contrary, the specific set considers the interaction of each reverse axis with each forward axis. Using the general rules, the equivalent rewritten location path has the length and can be computed in a time linear in the length of the initial location path and it adds as many identity-based joins as reverse steps are in the initial location path. Using the specific rules, the equivalent rewritten location path has the length and can be computed in a time exponential in the length of the initial location path, in the worst case, but does not add any extra joins.

Furthermore, the class of XPath 1.0 location paths with reverse axes that can not be rewritten with the above mentioned rules is syntactically characterised and it is shown how it can be rewritten using variables within XPath 2.0 or XQuery 1.0.

Location paths without reverse axes, as generated by the presented rewriting algorithm, enable efficient SAX-like streamed data processing of XPath.

Semantically encoding mathematics

Paul Libbrecht
paul@ags.uni-sb.de
Universität des Saarlandes
FR Informatik

Mathematical knowledge, like many, has traditionally been encoded in a visual fashion. The classical knowledge of a course is presented on a blackboard with oral comments, the classical presentation of a handbook or article is also a printed form, generally using the TeX family of systems. As any presentation language, re-usability is rather poor for such a language.

Mathematical knowledge, however, enjoys a rather good structure as mathematicians tend (and claim) to think in a logical fashion. As such the mathematical ontology can be made of theorems, definitions, examples and relations between them.

We discuss "semantic encodings" of mathematical content and present one candidate: the OMDoc language. In this language, mathematical formulas, a central part of the mathematical knowledge, are encoded "abstractly" using symbols which have a clearly identified meaning and can be defined in other parts of the document.

The OMDoc language has been extended for purpose of the ActiveMath environment which presents mathematical content in a web-browser in an adapted fashion. The major extensions pertain to the metadata informations stored about each of the individual items. This is put to good

use to choose the appropriate content to a user whose knowledge is previously estimated. The semantic encoding of the content is, itself, used to enrich the presented content with interactive facilities.

XML Data Integration using XPathLog and LoPiX

Wolfgang May
may@informatik.uni-freiburg.de
Universität Freiburg
Institut für Informatik

Due to many questions at the workshop, the talk had a second title, "From F-Logic to XPathLog": In addition to the presentation of XPathLog, also its development as a reconciliation between F-Logic as a "proprietary" data model and language for knowledge representation and data integration, and the standards of the XML world is described.

The XPathLog language is a Datalog-like extension of XPath for querying, manipulating and integrating XML data. Based on navigation and filtering, its basic, internal semantics is closely related to F-Logic. The querying part extends XPath with binding variables to XML nodes that are "traversed" when evaluating an XPath expression. Variables can be bound to literals, nodes, and even names, allowing for metadata reasoning. The variable bindings can be output as answers, or they can be communicated to the rule head for specifying updates in the database. In contrast to other approaches, the XPath syntax and semantics is also used for a declarative specification how the database should be updated: when used in rule heads, XPath filters are interpreted as specifications of elements and properties which should be added to the database.

The restriction that XML uses a tree data model directly effects the semantics of updates: if an update specifies that some subtree of a document should be inserted also at another place, the subtree must be copied. Thus, for references into this tree, it must be decided whether they point into the original or into the copy; "sharing" subtrees - as in graph data models such as OEM or F-Logic - is not possible. On the data integration level, when restructuring trees, fusing nodes, and introducing synonyms, this problem occurs even more.

Thus, as a data manipulation and integration language, XPathLog – and the LoPiX implementation – internally use a graph-based, edge-labeled model, called XTreeGraph. The XTreeGraph extends the basic XML data model by modeling multiple overlapping trees, and thus allows for restructuring existing XML trees into a densely connected graph database. XML result trees are then defined as XML tree views by projections from this database.

Thus, the "pure" XPathLog language provides an intuitive Datalog-style extension of XPath for querying, data manipulation, and data integration which is very close to the standard syntax and semantics of XPath. Extended features of the language provide a class hierarchy (including nonmonotonic inheritance), a lightweight signature formalism (which is also used for defining tree views from the XTreeGraph), and data-driven Web access to extend the database with further XML documents. The expressiveness and flexibility of the full language makes it a candidate for combined handling of data, schema-metadata, and semantical metadata such as ontologies.

Project homepage: <http://www.informatik.uni-freiburg.de/~may/lopix/>

ER Modeling and XML Modeling

Bernhard Thalheim
thalheim@informatik.tu-cottbus.de
BTU Cottbus
Institut für Informatik

It is commonly accepted that the dialects of XML (let us call them XML++) allow a logical representation of views of a large variety of users and applications. If we need cooperation or worse integration, a conceptual modeling language is a must. XML++ may be considered to be logical language (similar to the database slang that considers SQL to be a logical language). This conceptual language must meet a number of requirements such as well-foundedness, extensibility, view mechanisms, adaptable semantics, powerful mapping facilities, a theory of integration and - last but not least - a good development methodology. The higher-order entity-relationship modeling (HERM) approach has reached this maturity. It can be used for conceptual modeling of XML++. This approach has successfully been experienced during development of more than 30 large information-intensive internet sites. Therefore, HERM is THE conceptual modeling language for XML++. Even more, the pitfalls of the rule-triggering approach are omitted. HERM has a powerful theoretical basis.

W4

Well-founded Semantics for the World Wide Web

Carlos Damásio
cd@di.fct.unl.pt
Universidade Nova de Lisboa
Fac. Ciências e tecnologia

In this talk, I argue that well-founded semantics is an interesting basic approach for providing semantics to RuleML. It allows non-monotonic default negation and has polynomial data complexity. Moreover, it can handle both positive and negative recursion. The existence of an undefined truth-value is essential for representing non-terminating computations in the Semantic Web. I also presented extensions for reasoning in face of uncertainty and contradiction (paraconsistency), dynamically update knowledge bases and mechanisms for reactive behaviour. It is also described an architecture and algorithms for fully distributed tabled query evaluation, which terminates for every datalog query. The architecture is fully inter-operable and are described briefly algorithms for detecting termination and strongly connected components.

A Rule-based Language for Querying and Transforming XML Data

Sebastian Schaffert
schaffer@informatik.uni-muenchen.de
Ludwig-Maximilians-Universität München
Institut für Informatik

The growing importance of XML as a data interchange standard demands languages for data querying and transformation. Since the mid 90es, several such languages have been proposed that are inspired from functional languages (such as XSLT) and/or database query languages (such as XQuery). This talk addresses applying logic programming concepts and techniques to designing a declarative, rule-based query and transformation language for XML and semistructured data.

The talk first introduces issues specific to XML and semistructured data such as the necessity of flexible "query terms" and of "construct terms". Then, it is argued that logic programming concepts are particularly appropriate for a declarative query and transformation language for XML and semistructured data. Finally, a new form of unification, called "simulation unification", is proposed for answering "query terms".