Rules in the Semantic Web Services Language (SWSL): An Overview for Standardization Directions

Benjamin Grosof bgrosof@mit.edu
Massachusetts Institute of Technology, Sloan School of Management,
50 Memorial Drive, Cambridge, MA 02142, USA,
http://ebusiness.mit.edu/bgrosof

Michael Kifer kifer@cs.sunysb.edu
State University of New York at Stony Brook,
Dept. of Computer Science, Stony Brook, NY 11794-4400, USA,
http://www.cs.sunysb.edu/ kifer

David L. Martin martin@ai.sri.com
SRI International,
Room EK282, 333 Ravenswood Avenue, Menlo Park, CA 94025-3493, USA,
http://www.ai.sri.com/people/martin

Abstract

We give an overview of rules in the Semantic Web Services Language (SWSL) from the viewpoint of directions for standardization. This includes requirements, tasks about services, kinds of knowledge, combining rules with ontologies, suitability of fundamental knowledge representations and desirability of particular expressive features.

Note: To Find More about this paper

For updated/extended versions of this paper, and additional related material, see `http://ebusiness.mit.edu/bgrosof/#SWSLPosnPapForW3RuleWksh` starting in early- or mid-April 2005.

1. Intro to Semantic Web Services Language (SWSL)

The promise of Web services and the need for widely accepted standards enabling them are widely recognized, and considerable efforts are underway to define and evolve such standards in the commercial realm. Prominent among these are: Oasis's UDDI, BPEL4WS, and Web Services Security; and W3C's WSDL and Choreography Description Language.

At the same time, recognition is growing of the need for richer semantic specifications of Web services, based on a comprehensive representational framework that spans the full range of service-related concepts. Because an expressive representation framework permits the specification of many different aspects of services, it can provide a foundation for a broad range of activities, across the Web service lifecycle, while enabling radically more and cheaper *reuse* of such specification knowledge across those aspects, across applications and organizations, and over the duration of the lifecycle. It can support cheaper, broader, and deeper automation of many service tasks, including: service selection and invocation; translation of message content between heterogeneous interoperating services; service moni-

toring and recovery from failure; service authorization; contracting, negotiation, and supply chain management for services; and composition of services, along with their verification, simulation, and configuration.

The Semantic Web Services Language (SWSL), and the associated First-order Ontology for Web Services (FLOWS), have been developed to meet this need. SWSL is a product of the Semantic Web Services Initiative (SWSI), a collaborative international research – and early-phase standards – effort.

SWSL is a general-purpose logical language, with certain features to make it usable with the basic languages and infrastructure of the Web. These features include URIs, integration of XML built-in types, and XML-compatible namespace and import mechanisms. SWSL includes two aspects/layers of expressiveness: SWSL-FOL and SWSL-Rules. SWSL-FOL is a first-order logic, extended with features from HiLog and the frame syntax of F-logic. SWSL-Rules is a full-featured declarative logic programs (LP) language, which includes a novel combination of features from Courteous logic programs, HiLog, and F-logic.

SWSL includes a presentation syntax, nearly all the elements of which are common to both SWSL-FOL and SWSL-Rules, so as to promote the ability of developers to easily work with both, and to facilitate various kinds of interchange and interoperation between both. The presentation syntax is designed for readability, and incorporates a number of convenience features, such as an object-oriented style of presentation, which can be used to improve code organization and comprehensibility, but without changing the expressiveness and tractability of the underlying logical systems. An XML-based markup syntax, drawn from RuleML, is also specified.

Developed in conjunction with SWSL's language is FLOWS, an axiomatized ontology of service concepts, which provides the conceptual framework for describing and reasoning about services. FLOWS is specified in SWSL-FOL, and a partial expression of it is also specified in SWSL-Rules. FLOWS is built upon The Web Ontology Language for Services (OWL-S) and the Process Specification Language (PSL) as primary starting points.

In addition to SWSL and FLOWS, the SWSL effort also has developed some guidance as to how SWSL-Rules and SWSL-FOL can be used together, examples of their use, and a discussion of how FLOWS-based service descriptions can be grounded in the concrete descriptions of messages and protocols provided by WSDL.

Information about SWSL is available at:
`http://www.daml.org/services/swsl` and `http://www.swsi.org`. A substantial design report about SWSL is nearing completion and is planned for public release in mid-April 2005 on those sites. Several recent detailed presentation slidesets about SWSL's rules aspect are available, including:
`http://ebusiness.mit.edu/bgrosof/paps/talk-daml-rules-pi-2004-12.pdf` (section "SWSI Rules Update").

2. Two Fundamental KRs Meeting Requirements from Service Tasks, Ontologies

We begin by summarizing requirements drawn from various tasks about services and available kinds of ontologies, and how they motivate two distinct fundamental knowledge repre-

sentations (KR's). These motivations draw on an extensive requirements analysis performed by the SWSL Committee.

The overall SWSL language includes two distinct fundamental knowledge representations (KRs):

- declarative logic programs (SWSL-Rules); and
- first order classical logic (SWSL-FOL).

The SWSL-Rules KR is especially well suited to represent available knowledge and desired patterns of reasoning for several of the SWS tasks, including:

- authorization policies (for security, access control, confidentiality, privacy, and other kinds of trust);
- contracts (partial or complete, proposed or finalized);
- monitoring of processes to recognize and handle exceptions or other dynamic conditions (e.g., monitoring of performance of contracts to detect and respond to violations of contract provisions such as late delivery or non-payment);
- advertising, discovery, and matchmaking (e.g., advertisements and requests for quotation or requests for proposals can be regarded as partial contract proposals);
- semantic mediation, especially translation mappings that mediate between different ontologies or contexts and thus between knowledge expressed in those different ontologies or contexts (e.g., to translate from the output of one service to the input expected by another service); and
- object-oriented ontologies that use default inheritance with priorities and/or cancellation (e.g., in the manner of the Process Handbook (`http://ccs.mit.edu/ph`).

In particular, the capabilities of the Rules KR for logical nonmonotonicity (negation-as-failure and/or Courteous prioritized conflict handling) is used heavily in many use case scenarios for each of the above tasks and its associated kinds of knowledge.

The SWSL-FOL KR is especially well suited to represent available knowledge and desired patterns of reasoning for several others of the SWS tasks, especially revolving around the process model, including:

- composition of services, and associated planning using their process models;
- analysis, verification, and validation of services in terms of their process models; and
- ontologies expressed in first order classical logic, e.g., PSL or OWL-DL (Description Logic).

In particular, the capabilities of the SWSL-FOL KR for disjunction, reasoning by cases, contrapositive reasoning, and/or existentials are used heavily in many use case scenarios for each of the above tasks and its associated kinds of knowledge.

3. Combining LP/SWSL-Rules with FOL/SWSL-FOL

Issues arise in combining knowledge in nonmonotonic LP with knowledge in FOL; there is no known KR that supersumes both in their full generality. SWSL has adopted an approach that uses Horn LP, including Description Logic Programs as a special case, as a basic "bridge" KR to combine knowledge between/across these two distinct fundamental KRs. The approach is being extended by using results about hypermonotonic reasoning: this enables Courteous LP to be used as a much more expressive "bridge" KR.

4. Desirable KR Expressive Features

The SWSL effort has identified several KR expressive features as desirable based on the requirements analysis. Going beyond Horn, one very strongly desirable feature is negation-as-failure; this enables nonmonotonicity. Several other features are also strongly desirable because they provide major expressive conveniences, including: Courteous prioritized conflict handling, informational built-in predicates, HiLog, frame syntax, Lloyd-Topor "syntactic sugar", and unification equality.

SWSL has identified several other features which in their full generality raise issues that may require considerable further research in order to reach the mature degree of scientific understanding needed to underpin standardization near-term, including: reification, quotation, and user-defined equality.

5. SWSL's Relationships to RuleML and WSML

After evolving largely separately from RuleML (`http://www.ruleml.org`) for a while, SWSL then (since 2004) largely joined forces with RuleML. SWSL now shares its syntax and semantics with RuleML. The presentation syntax for both was drawn mainly from the SWSL effort and then was adopted by RuleML. The XML markup syntax for both was drawn mainly from the RuleML effort and then was adopted by SWSL. SWSL first developed in detail some expressive features that RuleML had not previously focused upon, including the Hilog feature. RuleML then, in collaboration with SWSL, has incorporated most of these. However, for now SWSL mainly has punted on representing procedural attachments for actions and tests/queries, which are heavily used in production rule systems and many other commercially important rule systems/applications, while RuleML enables such procedural attachments via its Situated feature (effecting and sensing).

SWSL also then (since 2004) started collaborating, more loosely, with the WSML (Web Services Mediation Language) effort as well. WSML takes a largely similar approach in many regards to RuleML and SWSL. There has been mutual influence between these three efforts/designs, earlier particularly from RuleML to SWSL and to WSML but more recently (especially since latter 2004) in all directions.

Web Services Modeling Ontology (WSMO) is a parallel effort to define an ontology and a language for Semantic Web services. Like SWSL-Rules, WSMO's rule language (WSML) is largely based on F-logic and these languages share much of the logical expression syntax. However, the two groups have pursued complementary goals.

6. Implementation Status; SweetRules Toolset Platform for RuleML

SWSL has not been directly implemented extensively yet. This is because the SWSL effort has to date focused mainly on requirements, specification, and use case scenario development. Once SWSL's design report is released in April 2005, SWSL plans to embark upon more ambitious implementation efforts, largely using tools for RuleML and F-Logic, including notably SweetRules and Flora-2.

RuleML – with which SWSL's rule language largely coincides in syntax and semantics – has already indeed been extensively implemented.

In particular, SweetRules V2 (`http://sweetrules.projects.semwebcentral.org`), released in Dec. 2004, is a uniquely powerful integrated set of tools for semantic web rules and ontologies. Moreover, it is open source. It has been developed by a multi-institutional effort involving about a dozen universities and companies, led by MIT Sloan. Implemented in Java, SweetRules has a compact codebase (currently about 20,000 lines of code total for several dozen tools).

SweetRules supports the powerful Situated Courteous Logic Programs extension of RuleML, including prioritized conflict handling and procedural attachments for actions and tests. SweetRules' capabilities include semantics-preserving translation and interoperability between a variety of rule and ontology languages (including XSB Prolog, Jess production rules, HP Jena-2, IBM CommonRules, OWL, and SWRL), highly scaleable backward and forward inferencing, and merging of rulebases/ontologies, as well as some analysis and authoring. The Description Logic Programs (DLP) KR subset of FOL, and the associated DLP-fusion technique, is supported via the capability to translate and merge DLP OWL ontologies into SCLP rulebases.

SweetRules thus implements a great deal of the expressiveness of SWSL Rules – and of the SWSL "bridging" approach to combining LP knowledge/rules with FOL/DL knowledge/ontologies.

SweetRules also implements, to a lesser extent, much of the expressiveness of SWSL FOL – via support of DLP OWL, Horn LP, SWRL (whose rules are a subset of FOL RuleML), and translation to KIF.

SweetRules' pluggability and composition capabilities enable new components to be added relatively quickly. SweetRules thus comprises a *platform* – moreover, the first such – for semantic web business rules. SweetRules currently enables powerful interoperability and inferencing across three of the four families of rule systems that are commercially most important today – production rules, relational databases, and Prolog – and thus across the applications and organizations that utilize these rule systems. The fourth such family, Event-Condition-Action rules, is closely related to production rules.

Flora-2, a backward-reasoning LP rule system developed at SUNY Stonybrook on top of XSB, provides a powerful implementation of Prolog that is extended with several additional KR features including Hilog and frame syntax cf. F-Logic, that are not yet supported in SweetRules but which are in full SWSL and RuleML. However, Flora-2 does not yet support the Courteous KR feature.

In current work, tool support for the SWSL/RuleML presentation syntax is being developed as part of SweetRules by the MIT Sloan group. Its release is planned for approximately summer 2005.