The SILK System: Scalable Higher-Order Defeasible Rules

Benjamin Grosof[1], Mike Dean[2], and Michael Kifer[3]

[1] Vulcan Inc., Seattle, Washington, USA, BenjaminG@vulcan.com
[2] BBN Technologies, Ann Arbor, Michigan, USA, mdean@bbn.com
[3] Stony Brook University, Stony Brook, New York, USA, kifer@cs.stonybrook.edu

Abstract. SILK is a new knowledge representation (KR) language and system that integrates and extends recent theoretical and implementation advances in semantic rules and ontologies. It addresses fundamental KR requirements for scaling the Semantic Web to large knowledge bases in science and business that answer questions, proactively supply info, and reason powerfully. SILK radically extends the KR power of W3C OWL RL, SPARQL, and RIF, as well as of SQL and production rules. It includes defaults (cf. Courteous LP), higher-order features (cf. HiLog), frame syntax (cf. F-Logic), external actions (cf. production rules), and sound interchange with the main existing forms of knowledge/data in the Semantic Web and deep Web. These features cope with knowledge quality and context, provide flexible meta-reasoning, and activate knowledge.

1 Overview

SILK[4] (Semantic Inferencing for Large Knowlege) is a new knowledge representation (KR) system that includes a language, reasoner, user interface, and interchange capabilities. It is a highly ambitious effort by Vulcan Inc., which also develops Semantic MediaWiki+[5] and venture invests in semantic technology. SILK addresses fundamental KR requirements for scaling the Semantic Web to widely-authored Very Large Knowledge Bases (VLKBs) in business and science that answer questions, proactively supply info, and reason powerfully. SILK is part of Vulcan's Project Halo[6].

SILK radically extends the KR power of W3C OWL 2 RL, SPARQL, and RIF — and of SQL and production rules. It includes:

- Defaults and robust conflict handling — *to cope with knowledge quality and context*;
- Higher-order and flexible meta-reasoning — *to elevate meta-data to meta-knowledge*; and

[4] http://silk.semwebcentral.org
[5] http://wiki.ontoprise.de
[6] http://projecthalo.com

– Actions and events, cf. production rules and process models — *to activate knowledge*.

SILK's language and semantics are based on *Hyper* Logic Programs, a new expressive extension of declarative logic programs (LP).

Its defaults and conflict handling features are based on:
(1) *Argumentation Theories* [1], a new expressive extension (and simplifying reformulation) of Courteous LP; and
(2) *Hypermonotonic mapping*, a new approach for sound interchange with first-order classical logic (FOL) that generalizes the method used in Description Logic Programs [2] and OWL 2 RL (i.e, OWL's Rules profile).

SILK's higher-order and meta-reasoning features are based on *HiLog* LP [3], a form of quasi higher-order that is expressively reducible to first-order, along with reification.

SILK also supports well-founded (nonmonotonic) negation, frame syntax cf. F-Logic [4], logical functions, integrity constraints, Lloyd-Topor transformations, loading and dynamic querying of remote Web knowledge sources, triggering of procedurally attached actions such as remote Web services, causal effects of actions, and a number of other features.

Despite its expressive power, Hyper LP is tractable under non-onerous restrictions; it has practically the same computational complexity as Horn LP (similar to Horn FOL).

The syntax and semantics of SILK have been heavily influenced by RuleML, Semantic Web Services Language (SWSL), FLORA-2[7], W3C Rule Interchange Format (RIF), and their predecessors.

The SILK effort includes development of use cases and requirements in e-science, business policies, ontology mapping, e-commerce, and biomedical.

Overall, SILK aims to redefine the KR playing field for the Semantic Web, business rules, and rule-based process management. Current commercially important systems there have barely progressed in expressive power since the early 1980's. SILK provides new techniques and implementation to integrate two decades of fundamental research advances in semantic rules and ontologies based on declarative logic programs, the same KR that underpins SQL and relational database management systems as well as SPARQL and OWL RL.

2 An Example

Figure 1 shows a SILK rule set that demonstrates several language features, including conflict resolution between two default rules, in an environmental process modeling domain.

The initial statement provide metadata about the ruleset (identified using the relative URI `<>` to refer to the containing document). These are followed by initial conditions in state 0 and a constraint. Squamish is a river in British Columbia.

[7] `http://flora.sourceforge.net`

A rule describing the effect of a toxic discharge event (labelled `tdf1`) is followed by a "frame" axiom (labelled `pef1`) that characterizes steady-state conditions. Use of the `silk:overrides` predicate in the statement labelled `pr1` indicates that `tdf1` has a higher priority than `pef1`, i.e. that only the head of `tdf1` is implied when the bodies of both `tdf1` and `pef1` are satisified.

The statement labelled `UhOh` adds a toxic discharge event and the final comment includes statements that are correctly inferred, using rules `pef1` and then `tdf1`.

```
<>[rdfs:comment->"Toxic discharge into a river causes fish die-off",
   owl:versionInfo->"$Id: silk-ruleml-2009-challenge.tex 880 2009-09-04 14:38:50Z mdean $",
   dc:source->"http://silk.semwebcentral.org/SemTech-2009-v31.pdf"] ;

// Initial facts, and an "exclusion" constraint that fish count has a unique value
occupies(trout, Squamish) ;
fishCount(0, Squamish, trout, 400) ;
!- fishCount(?s, ?r, ?f, ?C1) and fishCount(?s, ?r, ?f, ?C2) | ?C1 != ?C2 ;

// Action/event description that specifies causal change, i.e., effect on next state
@{tdf1} fishCount(?s+1, ?r, ?f, 0) :- occurs(?s, toxicDischarge, ?r) and occupies(?f, ?r) ;

// Persistence ("frame") axiom
@{pef1} fishCount(?s+1, ?r, ?f, ?p) :- fishCount(?s, ?r, ?f, ?p) ;

// Action effect axiom has higher priority than persistence axiom
@{pr1} silk:overrides(tdf1, pef1) ;

// An action instance occurs
@{UhOh} occurs(1, toxicDischarge, Squamish) ;

/* As desired:  |= fishCount(1, Squamish, trout, 400) and
                   fishCount(2, Squamish, trout, 0) ; */
```

Fig. 1. SILK rule set demonstrating conflict resolution and other features

3 Implementation

SILK Version 1 employs a version of FLORA-2 (known as Cornsilk) extended to support defaults and argumentation theories [1].

SILK Version 2 also uses Cornsilk as its reasoning engine but incorporates a Java API for language parsing, user interface functions (including a Jabber/XMPP Instant Messaging interface), interfaces to the Semantic Web and Web services, and interoperability using RIF. At ISWC 2009, Version 2 will be used to demonstrate prioritized conflict handling over Semantic Web data in a relevant application domain.

Figure 2 shows its current architecture. There, “xform” stands for “transformation” and dotted lines indicate future capabilities.

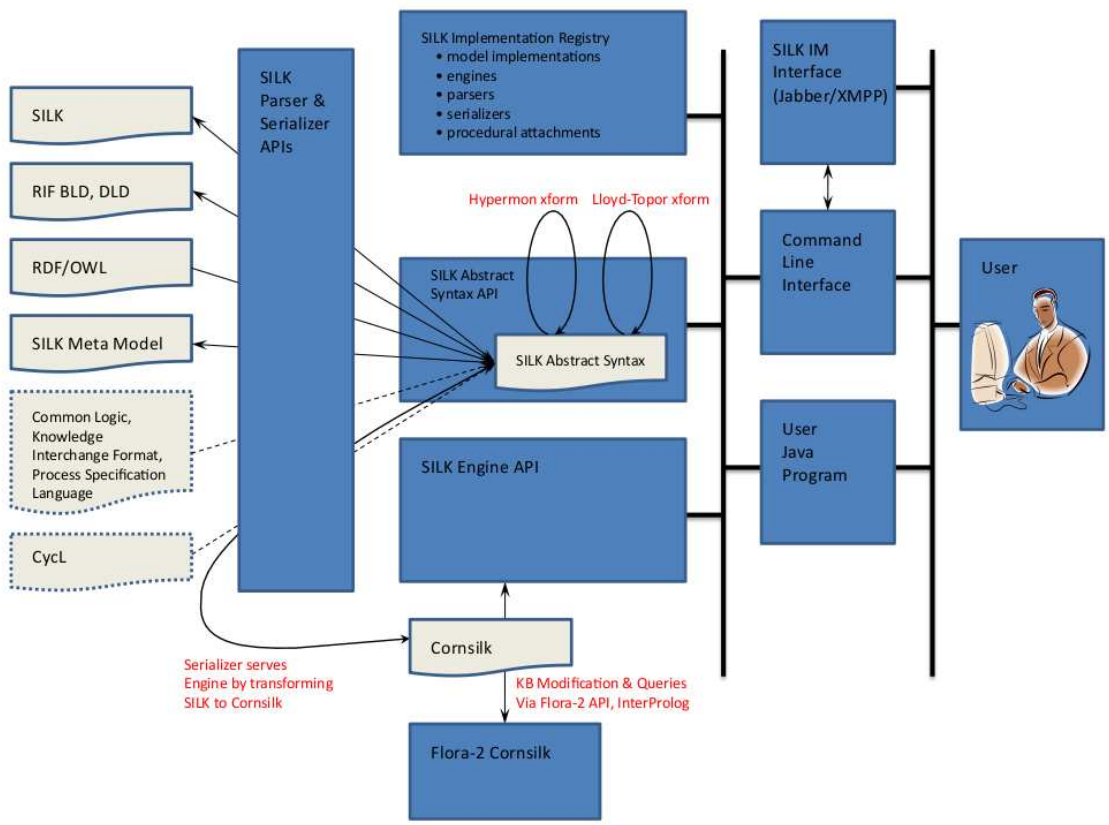

Fig. 2. SILK Version 2 System Architecture

A public release of SILK, free for research use, is planned for late 2009.

SILK Version 3 will continue to use the Java API and emphasize scalability and performance. More information can be found on the SILK website at the URL above.

4 Acknowledgments

SILK's development is sponsored and led by Vulcan Inc., in its Project Halo's Advanced Research effort (HalAR).

The SILK team includes over 30 people from 15 organizations. In addition to the authors, contributors to SILK include: Carl Andersen, Juergen Angele, Bert Bredeweg, Ken Barker, Mark Burstein, Vinay Chaudhri, Peter Clark, Bill Ferguson, Richard Fikes, Paul Fodor, Michael Gelfond, Keith Goolsbey, Mark Greaves, Dave Gunning, Paul Haley, Daniel Hansch, Bill Jarrold, Dave Kolas, Georg Lausen, Doug Lenat, Senlin Liang, David Martin, Sheila McIlraith, Ken Murray, Ben Rode, Matt Rubin, Jeff Sherman, Shirin Sohrabi, Terance Swift, Ray Tomlinson, Raphael Volz, Hui Wan, Jesse Wang, and Michael Wellman.

References

1. Wan, H., Grosof, B., Kifer, M., Fodor, P., Liang, S.: Logic programming with defaults and argumentation theories. In: Proc. 25th International Conference on Logic Programming (ICLP 2009), Pasadena, California (July 2009)
2. Grosof, B.N., Horrocks, I., Decker, S., Volz, R.: Description logic programs: Combining logic programs with description logic. In: Proc. 12th International Conference on the World Wide Web (WWW-2003), Budapest, Hungary (May 2003)
3. Chen, W., Kifer, M., Warren, D.: HiLog: A foundation for higher-order logic programming. Journal of Logic Programming **15**(3) (February 1993) 187–230
4. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. Journal of ACM **42** (July 1995) 741–843