A SILK Graphical UI for Defeasible Reasoning, with a Biology Causal Process Example*

Benjamin Grosof[1], Mark Burstein[2], Mike Dean[2], Carl Andersen[2], Brett Benyo[2], William Ferguson[2], Daniela Inclezan[3], and Richard Shapiro[2]

[1] Vulcan Inc., Seattle, Washington, USA, benjaming@vulcan.com
[2] Raytheon BBN Technologies, Cambridge, Massachusetts, USA, {burstein, mdean canderse, bbenyo, wferguson, rshapiro}@bbn.com
[3] SRI International, Menlo Park, USA, daniela.inclezan@ttu.edu

Abstract. SILK is an expressive Semantic Web rule language and system equipped with scalable reactive higher-order defaults. We present one of its latest novel features: a graphical user interface (GUI) for knowledge entry, query answering, and justification browsing that supports user specification and understanding of advanced courteous prioritized defeasible reasoning. We illustrate the use of the GUI in an example from college-level biology of modeling and reasoning about hierarchically-structured causal processes with interfering multiple causes.

1 Introduction to SILK

SILK[4] (Semantic Inferencing for Large Knowledge) is an expressive Semantic Web rule language and system equipped with scalable reactive higher-order defaults. The system includes capabilities for reasoning, knowledge interchange, and user interface (UI). Part of Project Halo[5], sponsored by Vulcan Inc., the SILK research program addresses fundamental knowledge representation (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. The SILK effort has over 15 contributing institutions, including Vulcan, Stony Brook University, Raytheon BBN Technologies, Cycorp, and SRI International.

SILK pushes the frontier of KR by combining expressiveness plus semantics plus scalability. It targets defeasibility, higher-order, and actions — including to support reasoning about complex processes that are described in terms of causality, hierarchical structure, and/or hypothetical scenarios. For example, reasoning about causal processes is a large portion of first-year college biology, often requiring multi-step causal chains and/or multiple grain sizes of description to answer a textbook or exam question. Longer-term, SILK targets widely collaborative KA by subject matter experts (SMEs), such as science students/teachers or business people, not just knowledge engineers (KEs) or programmers.

* This work is part of the SILK project sponsored by Vulcan Inc.

[4] http://silk.semwebcentral.org

[5] http://projecthalo.com

SILK has a new fundamental KR: *hyper* logic programs, which extends normal declarative logic programs (LP). Hyper LP is the first to tightly *combine* several key advanced expressive features: *defaults*, with strong negation and priorities, cf. courteous LP [1] with argumentation theories [2]; (quasi) *higher-order* syntax, reification, and meta-reasoning, cf. HiLog [3] and Common Logic; and procedural attachments to *external* actions (side-effectful), queries (to built-ins, web sources or services), and events (knowledge update flows), cf. situated/production LP [1] (and similar to production rules). KR languages supported for interchange include: SPARQL and RDF(S); SQL and ODBC (e.g., Excel spreadsheets); SILK, RIF (-BLD and -SILK), and OWL (-RL); Cyc (most of its KR and KB); and AURA [4]. AURA is a Project Halo system for question-answering in first-year college science and currently has a KB with tens of thousands of axioms about biology. AURA largely pre-dates SILK and employs a frame-based KR that is considerably less expressive than SILK.

Outline and Contributions: A previous version of SILK was presented in [5]. In the rest of this paper, we present a novel addition to SILK since then: a graphical user interface (GUI) for KA and querying that treats defeasibility.

2 SILK Graphical User Interface & Defeat Justifications

We have developed a graphical user interface (GUI) to the SILK system for knowledge entry, query answering, and justification browsing. The GUI is currently used by KEs and is being extended to support use by subject matter experts (SMEs). The GUI supports user specification and understanding of advanced courteous prioritized defeasible reasoning. It is implemented as a plug-in to the Eclipse Integrated Development Environment (IDE).

The GUI, pictured in Figure 1, offers users a number of capabilities. Entered SILK statements are syntactically validated and statement components (e.g., annotations) are color-coded for clarity. User debugging of rule bases is facilitated by automatic tracking of target queries' results against user changes to the rules. This also allows what-if explorations. The GUI also offers query result justification trees (technically, graphs) that can be explored incrementally, by expanding each tree node to display its children. At each node, the user can specify particular bindings to filter the portion of the justification tree that is displayed. Trees of this sort are also available for negative results (i.e., when a literal *cannot* be inferred), allowing developers to drill down and identify flaws in a desired chain of logical reasoning. This display mechanism also supports the reasoning chains found in courteous defaults by showing *defeated* ground rule instances — rules whose heads are not true, despite their bodies being true, due to conflict with other rules. Figure 1 shows an example of *refutation*-flavor defeat of a rule instance (and thus of its head atom A). The rule instance has a *candidate* argument — i.e., the rule's body is satisfied. But there also is a candidate *counter*argument (whose head is *neg* A) that has a *higher-priority* rule tag.

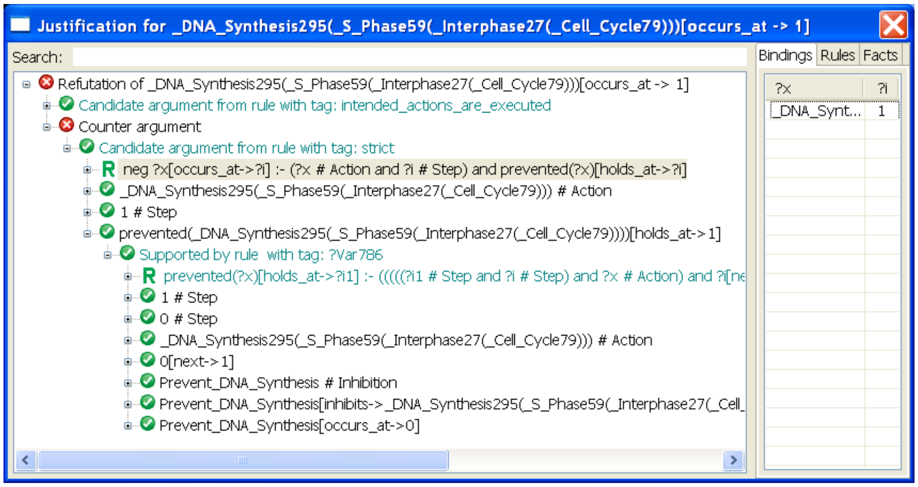

Fig. 1. SILK GUI: exploring the justification of defeat of a biology process rule

In (ASCII) **SILK syntax**, `a # c` means that `a` is an instance of class `c`. Skolems are prefixed by the underscore character ("_"). A courteous rule label term, used for prioritization, is called a *tag*.

Flexible control of selection and layout of items to display is achieved via running rules that are specified (e.g., by power users) in SILK itself.

To our knowledge, the SILK GUI is only the second justification exploration GUI for (prioritized) *defeasible* rules that have (declarative, model-theoretic) *semantics*. The first such system was DR-DEVICE [6], which displays defeat justifications, but with less extensive GUI functionality than SILK provides. Its Defeasible Logic KR is closely related to the courteous feature of hyper LP (see [7] for a comparison), but lacks higher-order and several other advanced expressive features of hyper LP.

3 Example: A Complex Causal Process in Biology

We have developed a novel approach to modeling and reasoning about hierarchically-structured causal processes, that smoothly handles interference/exceptions between multiple causes and elegantly treats the "frame problem" (inertia / persistence of causal fluents). It leverages hyper LP's prioritized defaults. To fully describe the approach is beyond the scope of this paper, however. Instead, we illustrate the approach with an example of college-level biology that shows the SILK GUI's novel capability to explore justifications in the presence of prioritized defeat.

In biology and medicine, a key process is the cell cycle in which a cell grows and then divides. (Control failure in this process causes cancer.) The cell cycle is a complex hierarchically-structured process. It consists of two phases (subprocesses): interphase and mitosis, in that temporal order. Interphase, in turn, consists of three subphases G1, S, and G2, in that order. Mitosis too has several subphases. Many of the above subphases in turn have sub-subphases, etc. DNA synthesis occurs during S phase, and indeed begins when S phase begins. This is the knowledge required to answer the following first-year college exam question[6]:

```
A researcher treats cells with a chemical that prevents
DNA synthesis from starting.  This treatment traps the cells
in which part of the cell cycle?
  Correct answer: G1.
```

That is, if DNA synthesis does not occur, then S Phase does not occur, and the cell cycle stops in the preceding phase which is G1.

Figure 1 shows a key intermediate step in SILK's inferencing — defeat of a candidate argument that: DNA Synthesis will occur for the question's focal cell cycle (`_Cell_Cycle79`), as is normal (i.e., "intended") in the cell cycle's process's cascade of causal phase steps. That argument is refuted because there is a higher-priority counter argument (itself undefeated) based on the preventive/inhibitory causal effect of the hypothetical scenario's chemical treatment.

4 Conclusions

A key direction in current and future SILK work is to increase SME friendliness of the UI in collaborative KA and querying, using in part controlled natural language. SILK is now being integrated with other portions of Project Halo, particularly AURA. We are refining a translation of Cyc biology etc. knowledge to SILK. See the SILK website for an **extended version of this paper**.

Acknowledgements: Thanks to the SILK team, esp. Paul V. Haley, Terrance Swift, Michael Kifer, David Gunning, Vinay Chaudhri, Michael Gelfond.

References

1. Grosof, B.N.: Representing E-Commerce Rules via Situated Courteous Logic Programs in RuleML. Electronic Commerce Research and Applications **3**(1) (2004)
2. Wan, H., Grosof, B., Kifer, M., et al.: Logic Programming with Defaults and Argumentation Theories. In: Proc. 25th Intl. Conference on Logic Programming. (2009)
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. Gunning, D., Chaudhri, V.K., et al.: Project Halo Update: Progress Toward Digital Aristotle. AI Magazine to appear, Fall 2010.
5. Grosof, B.N., Dean, M., Kifer, M.: The SILK System: Scalable and Expressive Semantic Rules. In: Posters and Demos, 8th Intl. Semantic Web Conf. (2009)
6. Bassiliades, N., et al.: Proof explanation in the DR-DEVICE system. In: Proc. 1st Intl. Conf. on Web Reasoning and Rule Systems. (2007)
7. Wan, H., Kifer, M., Grosof, B.: Defeasibility in Answer Set Programs via Argumentation Theories. In: Proc. 4th Intl. Conf. on Web Reasoning and Rule Systems. (2010)
8. Campbell, N.A., Reece, J.B.: Biology. 6th edn. Benjamin Cummings (2002)

[6] chapter 12 self-quiz question 15 in [8]