SAIC Advanced System Concepts

Authors:

- [Bradley Bebee](), SAIC/ASC
- [Bryan Thompson](), SAIC/ASC
- [Mike Personick](), SAIC/ASC
- [David Fado](), SAIC/ASC
- [Kendall Clark, Clark&Parsia]()

[SAIC]() Advanced System Concepts (ASC) researches, designs, and implements both prototype and production systems, primarily on behalf of government customers. These customers are typically but not exclusively members of the Intelligence Community, and their challenges typically lie within the area of knowledge management, knowledge discovery, and knowledge sharing.

ASC has made significant investments in Semantic Web and Web technologies, including RDF, OWL, SPARQL, lightweight (REST) web services, etc. In addition to the general need for interoperable solutions, ASC's government customers care especially about vendor and data independence, often because of regulatory requirements.

ASC's position is that *a standard for representing and exchanging rules and rule sets* -- and especially one that meshed well with existing standards like RDF, OWL, and "web architecture" -- *is an important next step, not only for the development of the Semantic Web, but also for the customers with which ASC is engaged.*

In what follows we focus on describing ASC use cases of and requirements for a **specification for publishing and exchanging rules and rule sets, both on the Web and in concert with OWL, RDF, and SPARQL** (hereafter, a "rules spec").

Use Cases

Federation and Semantic Alignment

The Problem. Analysis across distinct, separate knowledge bases. They often have to be distinct for legislative and other policy reasons. These separations are not clean and are often fractured. And there's no reason to believe that any particular KB is authoritative or complete with respect to the whole knowledge domain. Finally, each knowlege base has both an independent history and provenance, but also, often, independent schemas and underlying implementation techniques.

The Goal. To discover implicit knowledge, particularly links between actors, groups, and institutions, that exists across the totality of disparate knowledge bases. That is. to achieve a federated and, thus, a complete (or, as complete as is possible to have, given the state of knowledge at any point in time) view onto all the relevant knowledge.

These problems require data federation and semantic alignment -- by the former we mean some virtualized or realized view onto or of or across *all* the relevant knowledge bases; or, in other cases, across some subset of those knowledge bases. The goal of federation is to allow analysts to gain a clearer understanding of a problem domain (for example, terrorist threats) by looking at sets of partially

overlapping data sources, each with relevant but incomplete information. If the concepts and instances are aligned -- that is, semantically coherent across different knowledge bases, both in terms of schemas but also in terms of the properties of individuals -- then a lot of the distracting labor of dealing with the different syntax, schemas, and semantics of the different sources is ameliorated.

Thus, OWL's `sameAs` and `equivalentTo` are useful; but it's often the case that we need to do federation and alignment between RDF and non-RDF knowledge bases. In those cases we have to do "property transfers", that is, aligning transformations in which a property `foo:address` should be broken into `bar:street, bar:state, bar:postalCode`. That is, **we need to be able to do rule-like transformations between non-RDF systems in which making `OWL:sameAs` and `OWL:equivalentTo` assertions is not possible.**

Further, these federation-and-alignment scenarios are very dynamic, which is to say that they're often task-specific and goal-relative. A rule that makes sense in one scenario is irrelevant in another. For example, access to different knowledge bases can be a matter of policy and regulatory intent. Thus, **we need to express multiple, dynamic merger-and-alignment policies on the same data, often in the context of orthogonal access control policies.**

These knowledge bases are dynamic and may be updated frequently. Rules and interoperable representations of policy are vital to handling the issues raised by update. In short, this is a policy-rich environment:

1. analysts may only look at the data they are allowed to look at
2. in some cases, for the data which an analyst may access, the analyst is not allowed to modify the data in any way; in other cases, the analyst may be allowed to modify the data in the federated knowledge base, but not in the source knowledge base; in yet other cases, the analyst may be allowed to modify the source knowledge base, too.
3. analysts must not do anything that would prevent them from receiving updates of federated data from the original sources.

In cases where the source knowledge base cannot be updated, updated or new assertions may be placed into a separate knowledge base, and eventually re-merged with the federated knowledge base. The rules about which assertions are to be trusted most, by whom, and in what circumstances, are complex and need to be shared between various organizations and systems. A common representation and exchange format for rules would be of invaluable assistance in building and maintaining such complex, federated systems. For example, we need to be able to write rules to do RDF knowledge base maintenance, data cleansing & scrubbing, and business-logic validation. In short, *we want to use rules for RDF knowledge bases in a way analogous to the way XSLT and Schematron rules are used to maintain and validate XML instances.*

Agent Infrastructures

A second class of rules use cases for ASC is agent infrastructures. Another ASC project for a government customer is built around agent technology, particularly as a way of managing the interactions between users and information.

An open, expressive rules exchange format, which worked across disparate agent frameworks, would be very useful in helping relate agent to non-agent infrastructures, as well as allowing end-users to shape or influence agent behavior by writing new rules.

Conclusion

ASC's position about a W3C effort to create a rules spec is that the effort should make possible the exchange of rules and rule sets; that the semantics of this representation should be such that it meshes well with both RDF and OWL; and that the focus of such an effort should be less on legacy integration and more on creating a representation of rules for and on the Web.