Basis for Executable Open Vocabulary English as a Candidate Solution for CDSI

Note for the Cross Domain Semantic Interoperability Working Group 20061115

Adrian Walker Reengineering adriandwalker@gmail.com

Jim has set up an excellent framework for our discussions. As I understand it, the framework asks us to focus on the problem of interoperability, and to come up with candidate solutions. The only limit he imposes on a candidate is that, for interoperation of N stovepipe software systems, there should be no more than 2N adapters. This rules out candidates that would require N**2 adapters, one for each pair of stovepipes.

As covered in our opening discussion on Nov 13th, most if not all, of the candidates so far are ontologies in technical notations such as OWL. It is a feature of such ontologies that they contain comments in free English that say what the authors intended. However, since these ***are*** comments, they are thrown away by any software that uses the ontologies. Thus, the semantic intent of the authors is lost at run time, and cannot be easily communicated to the user community.

As mentioned in our meeting on Nov 13th, there is some emerging technology that supports a representation shift in the way that we write down and use knowledge for 2N interoperability. The general idea is that knowledge authors can work in a kind of Wiki environment, writing their knowledge into their browsers in open vocabulary, executable English. The technology is live, online [1,2]. Shared use is free.

To see why it may be a good idea to shift to this more human level of representation, consider the process of trying to make stovepipes interoperate using only lower level, technical notations. Typically, a team will assemble around a (virtual) whiteboard, and will discuss in English (or another human language) the meanings of the respective stovepipe concepts, and how to map them back and forth. A 2N consensus is arrived at, and is mapped into an ontology in a technical notation, with English comments. The ontology is then used as a basis for writing adapters for actual interoperation. When this software reaches users, the only recourse they have in the face of a counter-intuitive output is to look at the comments in the ontology and to try to figure out whether the intent was correctly mapped into say, raw OWL.

If, on the other hand, the team inputs its collective knowledge in executable English into browsers, there are several advantages. Knowledge can be input and tested by running it over sample data, in a tight edit-run loop, without getting into details at the programming level. Then, the supporting technology can carry through the semantic intent of the authors all the way to the user level. In particular, any output of the system can be explained, step-by-step, in English, at the level of a non-technical business user.

For simple examples of this, please see [3,4], and please also run the examples and look at the explanations. (There's nothing to install, just point a browser to [2]).

While this approach may seem radical, it actually complements [5,6,7] current work on technical ontology notations such as OWL and RDF while integrating two extra kinds of semantics [8,9]. It also appears promising for scalable interoperation among SQL databases [10], and for interoperation between SQL and RDF [5].

In summary, we can capture more interoperation semantics, and deploy it immediately, using emerging technology to write down knowledge for interoperation in open vocabulary, executable English. The candidate technology for this is live online, and is free for shared use by anyone with a browser.

References

[1] ww.reengineeringllc.com/A_Wiki_for_Business_Rules_in_Open_Vocabulary_Executable_English.pdf

[2] Internet Business Logic system. Online at www.reengineeringllc.com

[3] www.reengineeringllc.com/demo_agents/OntologyInterop2.agent

[4] www.reengineeringllc.com/demo_agents/SemanticResolution1.agent

[5] www.reengineeringllc.com/demo_agents/RDFQueryLangComparison1.agent

[6] www.reengineeringllc.com/demo_agents/OwlTest1.agent

[7] www.reengineeringllc.com/demo_agents/FeaReferenceModelOntology2.agent

[8] www.semantic-conference.com/program/sessions/S2.html

[9] Backchain Iteration: Towards a Practical Inference Method that is Simple
Enough to be Proved Terminating, Sound and Complete.
Journal of Automated Reasoning, 11:1-22

[10] www.reengineeringllc.com/Oil_Industry_Supply_Chain_by_Kowalski_and_Walker.pdf