Proposed Functional-Style Extensions for Semantic Web Service Composition

Barry Norton

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, S1 4DP Sheffield, UK
B.Norton@dcs.shef.ac.uk

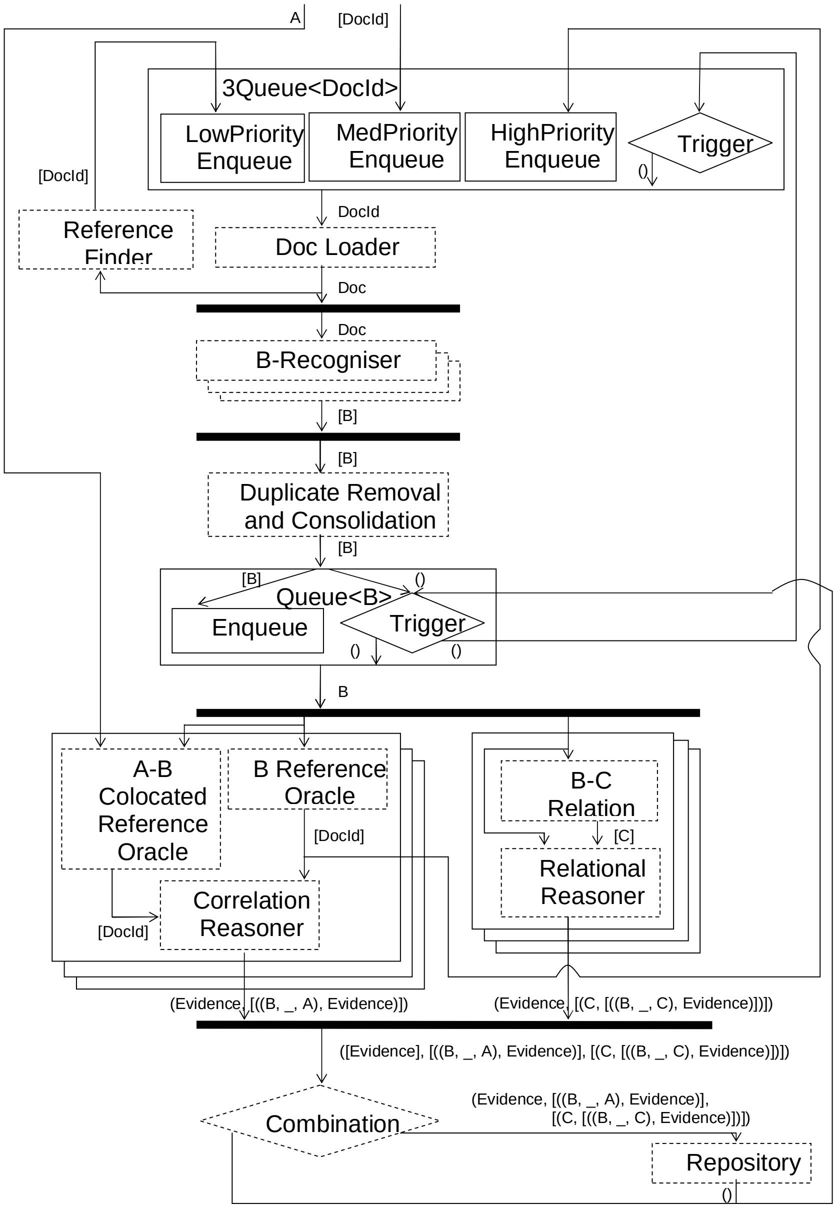
In a related paper [9] we set out how various parts of a semantic web service-based architecture for Armadillo[2], a harvesting tool for semantic annotation, can be instantiated with information extraction and related language services. We have constructed this as a workflow, illustrated as a whole on the following page, in BPEL4WS [3], reasoning, as have several other authors [7], that even while we should like to take advantage of semantic web service technology there exist few, if any, generally available choreography solutions for OWL-S. As a result we plan to take the lessons learned as input to an effort to implement and extend a 'coordination engine' for OWL-S in the CASheW-s project[1].

The broad goals are to implement an engine in the programming language Haskell [6] to which a workflow, expressed in an extended XML-encoded version of OWL-S, can be communicated (over SOAP). This workflow will then be converted to a process-algebraic representation in CaSE [10], a qualitatively-timed CCS derivative, from which coordination engines have already been formed for Microsoft COM via H/Direct [4]. Our plan is to use the GXS module of the HAIFA framework [5] to allow the same kind of binding of SOAP services.

Our existing generalised dataflow model in CaSE [10] allows a semantics for dataflows with loops, two loops can be seen over, and non-deterministic agents, shown over as flow-graph style diamonds. Both of these features are disallowed from the 'Flow' construct in BPEL (to construct the flow shown over it is necessary to use more than one workflow, BPEL being insufficiently algebraic), and the former syntactically restricted out of OWL-S, but very useful.

Our other major observation from BPEL was the frustrating need for use of its extended XPATH language together with mutable variables - anathema to the kind of analyses we should like to allow over designs. We have previously proposed parametrically polymorphic functional language-like features as a more suitable means to carry out transformation of XML-encoded data [8].

As well as allowing the binding of pure-functional Haskell operations at the 'mediation' level between services (as shown for instance in the second queue with $[B]$, a list of elements typed B, being transformed into (), the singleton type), we should like to go further in adding operations to workflow primitives. Firstly, we should like to bind Haskell functions to 'If-Then-Else' constructs, allowing us to form an operation like 'Trigger' (in the two Queue services) directly from a WSDL-described service without non-deterministic outputs. Furthermore, we should like to bind functions to the 'Split' and 'Join' primitives so that information can be propagated forward and consolidated through these features without the use of global variables, as shown in the diagram.



Acknowledgements

This work was carried out within the AKT project (http://www.aktors.org), sponsored by the UK Engineering and Physical Sciences Research Council (grant GR/N15764/01), and the Dot.Kom project, sponsored by the EU IST asp part of Framework V (grant IST-2001-34038). Also to be acknowledged are the students working on the CASheW-s project: Ravish Bhagdev, Xian Liu, Atheesh Sanka, Andrew Hughes and Simon Foster whose HAIFA project is being continued here.

References

1. CASheW-s engine project. http://savannah.nongnu.org/projects/CASheW-s-engine.
2. Fabio Ciravegna, Sam Chapman, Alexiei Dingli, and Yorick Wilks. Learning to harvest information for the semantic web. In *Proceedings of the First European Semantic Web Symposium*, May 2004.
3. IBM *et al.* Business process execution language for web services version 1.1. http://www-128.ibm.com/developerworks/library/ws-bpel, 2003.
4. Sigbjorn Finne, Daan Leijen, Erik Meijer, and Simon Peyton Jones. H/Direct: A binary foreign language interface for Haskell. In *Proc. 3rd ACM SIGPLAN International Conference on Functional Programming (ICFP-98)*, ACM SIGPLAN Notices. ACM Press, 1998.
5. Simon Foster. HAIFA : An interoperability framework for Haskell. MEng Dissertation, Department of Computer Science, University of Sheffield, 2004. http://www.dcs.shef.ac.uk/teaching/eproj/ug2004/abs/u1sf.htm.
6. Simon Peyton Jones. *Haskell 98 Language and Libraries.* Cambridge University Press, 2003.
7. David J. Mandell and Sheila A. McIlraith. Adapting BPEL4WS for the Semantic Web: The bottom-up approach to web service interoperation. In *Proc. 2nd Intl. Semantic Web Conference (ISWC2003)*, 2003.
8. Barry Norton. Eclipse as a development platform for semantic web services. Eclipse Technology Exchange (eTX04), 18th European Conference on Object-Oriented Programming (ECOOP-2004), 2004.
http://www.dcs.shef.ac.uk/∼barry/CASheW-s/Norton04.pdf.
9. Barry Norton, Sam Chapman, and Fabio Ciravegna. Developing a service-oriented architecture to harvest information for the semantic web. In *Proc. 1st AKT Workshop on Semantic Web Services.*
10. Barry Norton, Gerald Luettgen, and Michael Mendler. A compositional semantic theory for synchronous component-based design. In *Proc. 14th Intl. Conference on Concurrency Theory (CONCUR'03)*, volume 2761 of *LNCS*, 2003.