

MOSEW: A Tool Suite for Service Enabled Workflow

Mostafijur Rahman and Wendy MacCaull

Department of Mathematics, Statistics and Computer Science,
St. Francis Xavier University, Antigonish, NS, Canada
{x2013ici, wmaccaul}@stfx.ca

Abstract. Recently our research group introduced the notion of Service Enabled Workflow (SEW) with the integration of Semantic Web Service (SWS) and Workflow. In this paper, we present a Service Oriented Architecture (SOA)-based integrated software tool suite called MOSEW that provides functionalities to design and develop ontology based Quality of Service (QoS) aware SEWs.

1 Introduction

Service Enabled Workflow (SEW) [1] is relatively a new concept in the area of Semantic Web-based research. SEW considers workflow as a collection of tasks with specific control flow where tasks are carried out as services. While it has a lot of potential, SEW still requires a great deal of maturity and support of tools to become an industry standard. The MOSEW tool suite provides functionalities to design and develop ontology based SEWs where QoS-aware SWSs are discovered, selected and executed dynamically by a mobile agent for some of the tasks in a workflow to complete the overall execution.

2 QoS aware SWS Discovery, Selection and Execution

To manage the QoS specifications of the services effectively and utilize them to improve the service discovery approach, we designed a QoS conceptual model and integrated it into the OWL-S 1.2 framework. To read the OWL-S 1.2 service descriptions and execute the WSDL [2] service grounding, motivated by the efforts [2] and [4], we designed and developed the OWL API based OWL-S API that provides a Java API. The ontology based core matching algorithm, which extends algorithm [3] consists of two parts: basic functional (I/O) property-based matching and non-functional property (QoS)-based matching. We developed a ranking algorithm and placed it on top of the Service Discovery Engine that executes the semantic matchmaking algorithm. To access and execute the service, we used the grounding information of an OWL-S service.

3 MOSEW Architecture

We used this QoS-aware web service discovery, selection and execution approach in the Service Discovery and Execution engines of the MOSEW tool suite. The tool suite allows consumers to graphically define specifications of workflow tasks using ontology guided user interfaces and execute the workflows dynamically by a smart phone based software agent. Fig 1 shows the MOSEW architecture.

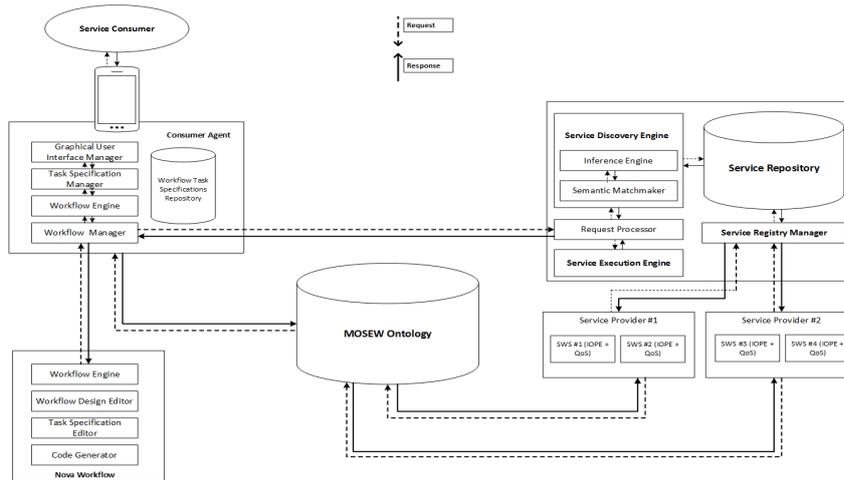


Fig. 1. MOSEW Architecture

4 Conclusion and Future Work

In this paper, we present MOSEW, a SOA-based integrated software tool suite that is used to design and develop SEWs running on mobile devices. We achieved this functionality through SWS discovery, selection, execution and semi-automatic run time composition. This type of service composition is time consuming and less flexible. The automatic service composition method generates the process model automatically or locates the correct services if an abstract process model is presented. In future, we will extend the MOSEW tool suite to support automatic service composition.

References

1. Altaf Hussain and Wendy MacCaull, Context Aware Service Discovery and Service Enabled Workflow, pp. 45-48. Canadian Semantic Web Symposium, 2013.
2. Evren Sirin and Bijan Parsa, The OWL-S Java API. Alternate Paper Tracks, 2003.
3. Massimo Paolucci et al. Semantic Matching of Web Services Capabilities, pp. 333-347. International Semantic Web Conference, 2002.
4. OWL-S API, <http://projects.semwebcentral.org/projects/owl-s-api/>