Specifying Workflows in SORASCS to Automate and Share Common HSCB Processes.

David Garlan, Bradley Schmerl, Vishal Dwivedi, Michael W. Bigrigg, Kathleen Carley
Carnegie Mellon University

ABSTRACT

SORASCS is a service-oriented-plus architecture that supports HSCB users in data processing, analysis, simulation, and reporting. A core need of HSCB users is to repeat common processes, for example when new data becomes available, or when attention shifts from one region of interest to another. We describe how SWiFT can be used to create, test, share, and modify SORASCS workflows. SWiFT is a web application that allows users to create workflows, distributes them in SORASCS for processing, allows the user to pause the workflow to examine intermediate results, and supports additional workflow analysis. This capability supports model re-use, rapid assessment, and improved tracking to support validation.

PRIMARY TRACK

Hybrid Models.

SECONDARY TRACK

Analytic Methods and Science and Technology.

DESCRIPTION

Today's data collection and analysis tool suites have a number of limitations that seriously restrict their ability to address the goals of HSCB. While there is a wide variety of tools and datasets supporting HSCB modeling and analysis, with more being created every day, most analysis ensembles handle a relatively limited set of model types and input sources. They are wired together with special-purpose, tool-specific and ad hoc integration code, making them brittle and requiring low-level systems expertise to reconfigure them in new ways or add new capabilities to the system. Even using existing configurations of tools requires deep and detailed knowledge of the idiosyncrasies of each tool. In other cases, functionality may be prepackaged for ease of use, but extending the capabilities of those tools involves considerable engineering and long delays. Moreover, it is difficult for analysts to replay previous analyses (perhaps on slightly different data), determine if some new form of analysis is even feasible (given the types of data and tool input/output requirements), or evaluate whether a desired analysis be completed in an acceptable amount of time.

SORASCS [1] provides a service-oriented architecture (SOA) platform for the HSCB community that allows a mixture of tools and their constituent capabilities, to be used in a distributed setting as web services. Such an environment supports the integration of a wide variety of heterogeneous model types, modeling and analysis tools, datasets, reports and simulations through a shared communication infrastructure, standards for data exchange, and common protocols for service invocation and execution.

SOA technologies also provide a way to compose multiple web services into enterprise-wide compositions through *orchestration*. However, using standard SOA mechanisms significant technical expertise is still required to assemble these services. For example, the Business Process

Execution Language (BPEL) standard [2] defines an orchestration language and is executable on most SOA platforms, but it is too low-level for non-experts to create, understand, or modify. For example, authors of BPEL orchestrations must understand details of how the data is passed between services, the protocols used to interact with services, and how to handle errors.

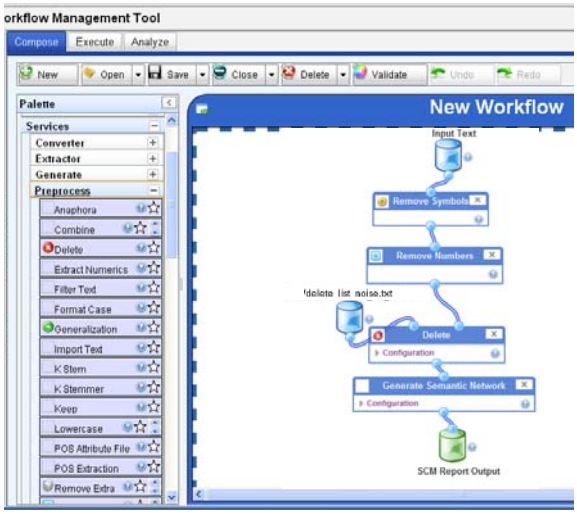

Figure 1. SWiFT Workflow Creation.

The SORASCS Workflow Tool (SWiFT) is a web-based tool that provides a high level of abstraction for defining HSCB workflows for transforming raw data into models. Workflows are defined as sequences of service calls, with data flowing between them. These are automatically translated by SWiFT into standard SOA orchestrations that can be deployed and executed in a distributed environment. Using services provided by multiple service providers via SORASCS,. SWiFT has the following features:

- **Workflow Creation.** Workflows are created as a sequences of services, which may be realized by primitive SORASCS services, by standalone applications, or by packaged workflows. Data typing on service ports ensures that correct workflows are constructed. When mismatches are detectedor SWiFT can locate transformation services to transform data of one type into another. Figure 1 illustrates the interface to SWiFT, where services are displayed through a web browser on a palette to the left, and may be dragged to and connected into a workflow on the right. By focusing an analyst on the steps involved in processing data, and not on technical assembly details, new workflows can be quickly assembled and tested. Previously defined workflows can be repeated on new or updated sets of data.
- **Workflow Sharing.** SWiFT workflows are translated into web services (defined by BPEL scripts) that are registered with SORASCS. This means that authors of workflows can publish these workflows with SORASCS and other analysts can find and reuse them. Furthermore, analysts can use SWiFT to tailor workflows to their own needs by providing predefined parameters, or by editing the workflow to meet their needs.
- **Workflow Execution Management.** Workflows can be executed to completion in SWiFT, which shows the progress of the execution. For debugging, analysts may define points where the workflow can be paused, allowing analysts to inspect whether workflows under development are doing what they expect. Intermediary data can be marked for inspection and persistent storage, allowing traceability of data after the workflow has executed, so that at any time an analyst can determine in detail what effects the workflow has had on data, and how conclusions were reached.
- **Workflow Analysis.** SWiFT provides a plug-in framework that allows it to be extended with workflow analysis tools that can aid the analyst in better understanding workflows. Such analysis plug-ins include advice about the ordering of steps in the workflow, performance predictions, an analysis of the security aspects of the workflow, and so on.

To achieve the above features, SWiFT is architected as a two-layer system. The top layer is a web-based front end that can be run in standard browsers. We use open-source libraries for this user interface that follow the mashup metaphor of Yahoo! Pipes[3], a common tool for combining and processing RSS feeds. SWiFT workflows extend this metaphor to include data transformation, stronger typing for data, and HSCB-relevant services.

The second layer is the SORASCS SOA-based platform. A SWiFT bridging service handles analysis, translation, and execution of SWiFT workflows, mapping them from the first layer to the second. Specifically, workflows are translated into BPEL and deployed on the Apache ODE open source BPEL engine in SORASCS. The use of BPEL as the underlying execution engine means that in the rare case where an analyst who understands BPEL and requires more complex orchestration of services, may use SOA standards to construct those workflows. These workflows can then also be used in conjunction with other workflows and services in SORASCS, maintaining a flexible, standards-based environment for HSCB analysis.

ACKNOWLEDGEMENTS

This work was supported in part by the Office of Naval Research - ONR -N000140811223. Additional support was provided by the center for Computational Analysis of Social and Organizational Systems (CASOS). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, or the U.S. government. We thank the following people who helped to implement SWiFT: Aparup Banerjee, Laura Glendenning, Yue Lu, Mai Nakayama, Nina Patel, and Hector Rosas.

BIOGRAPHY

David Garlan is a Professor of Computer Science and Director of Software Engineering Professional Programs in the School of Computer Science at Carnegie Mellon University. He received his Ph.D. from Carnegie Mellon in 1987 and worked as a software architect in industry between 1987 and 1990. He is considered to be one of the founders of the field of software architecture, and, in particular formal representation and analysis of architectural designs. In 2005 he received a Stevens Award Citation for fundamental contributions to the development and understanding of software architecture as a discipline in software engineering.

REFERENCES

[1] Garlan, D., Carley, K.C., Schmerl, B., Bigrigg, M., and Celiku, O. Using Service-Oriented Architectures for Socio-Cultural Analysis. In *Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE2009)*, Boston, USA, 2009.

[2] Dholaokia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S., Andrews, T., and Curbera, F. Business Process Execution Language for Web Services (BPEL4WS), Version 1.1. Technical Report, BEA, IBM, Microsoft, SAP, Siebel Systems, 2003.

[3] http://pipes.yahoo.com