RESTful Open Workflows for Data Provenance and Reuse

Kai Eckert* Dominique Ritze* Konstantin Baierer† Christian Bizer*

*Research Group Data and Web Science, University of Mannheim, Germany
{kai,dominique,chris}@informatik.uni-mannheim.de

†Berlin School of Library and Information Science, Humboldt University Berlin, Germany
konstantin.baierer@ibi.hu-berlin.de

ABSTRACT

In this paper, we present a workflow model together with an implementation following the Linked Data principles and the principles for RESTful web services. By means of RDF-based specifications of web services, workflows, and runtime information, we establish a full provenance chain for all resources created within these workflows.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Information ServicesWeb-based services

Keywords

Workflow;RESTful Web Services;Linked Data; Provenance

1. OPEN WORKFLOWS

We combine Linked Data and REST principles to create a workflow system based on the Web as underlying architecture – to communicate between stages of a workflow, to access resources and to provide new resources as results of workflows. Referring to the REST principles, everything in such an *open workflow* is a resource. Besides the entities that are consumed, transformed or created within a workflow, this holds in particular for the workflows themselves, the services and the communication artifacts controlling the workflow. These stateful resources form the provenance chain of the processed data resource. We use the term open workflow to distinguish such a web-based workflow from common workflow systems that are executed in a closed environment (cf. [7]). While service-oriented architectures focus on the reusability of services, open workflows additionally enable the reuse of the processed data, not only to reproduce results, but also to allow the branching of workflows to process intermediate results in new ways. A workflow is the composition of several steps to generate some output by processing some input. In the context of the Web, the single processing steps are web services. From the workflow perspective, a web service is seen as an atomic activity whose internals are hidden from the workflow system. The whole workflow ontology is structured into three different areas: specification, composition and execution.

Specification. Besides descriptive elements like a name, a web service is mainly characterized by its input and output parameters. A parameter is described by its type and can be either required or not. For convenience, a default value can be specified.

Composition. Within a workflow, web services are assigned to positions, cf. Figure 1. Like web services, a workflow has input ($param1$, $param2$) and output parameters ($param3$). Connectors (pink arrows) are used to connect resources to input parameters. For example, the output of $WS1$ on $Position1$ is the input of $WS2$ on $Position2$. The implementation of the workflow engine uses the type information of the parameters to verify the compatibility of the connections.

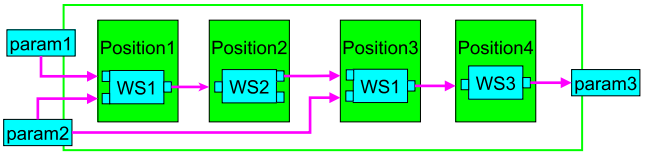

Figure 1: Example of a Workflow Procedure

Execution. Runtime information is represented by configurations, which contain all assignments of values to parameters of a web service or a workflow. A service is started by `POST`ing the URI of a configuration to it. Each configuration is persisted and exposed as Linked Data. When a web service or workflow is invoked, a job resource is created that asynchronously provides information about the status of the job and points to the result upon completion.

Figure 2 provides an overview of the whole ontology. Blue classes represent all parts of the specification, green ones the composition and white ones the execution part. The full ontology supports the publication of interoperable provenance information by using W3C PROV.

2. REST API

Following the Linked Data principles, all resources described by our ontology can be dereferenced to `GET` an RDF representation, including links to other resources. Figure 3 illustrates the execution of a web service: First, the client creates a configuration containing parameter assignments, persists them by means of a `POST` request to the

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577347 .

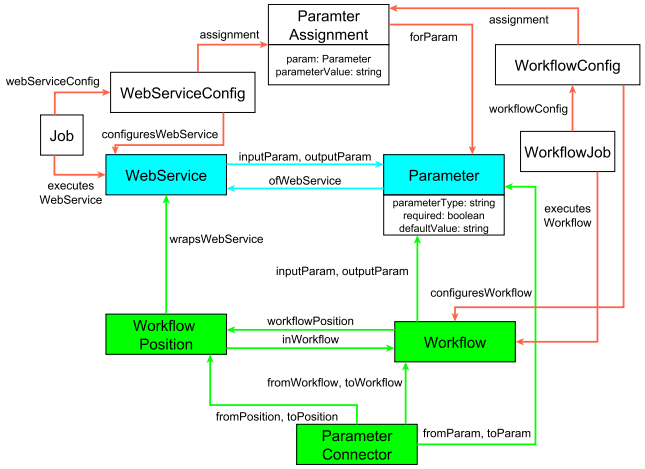

Figure 2: Workflow Ontology

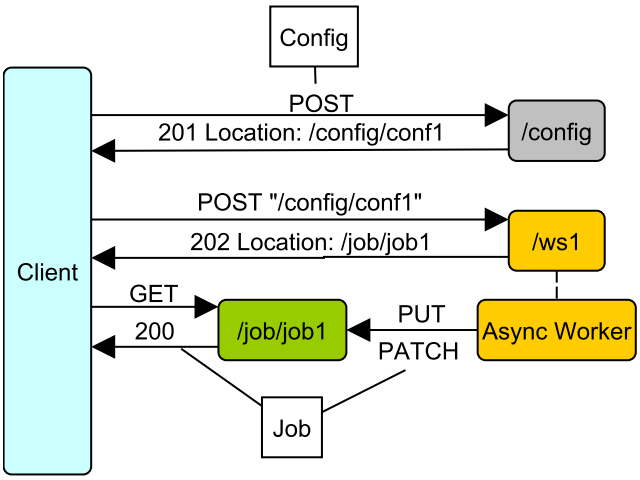

Figure 3: HTTP communication of a web service execution

configuration service and receives the URI of this configuration. It then `POST`s the configuration URI to the service which kicks off the asynchronous execution of the web service in the background but instantly returns the URI of a job resource. The asynchronous worker updates the job while the client polls the current status of the job. Workflows are executed in the same way: a workflow service consumes a configuration with the workflow input parameters and creates web service configurations according to the description.

3. IMPLEMENTATION

The implementation, called *Omnom*,[1] consists of a lightweight browser-based user interface and a backend that implements the workflow execution engine [2]. The backend is written in Java, using JAX-RS for inter-service communication and as API for the GUI. The user interface is a single-page application with Drag and Drop support. It is based on BackboneJS and Bootstrap, allowing users to upload files, create and execute workflows, and monitor the resulting jobs.

[1]Demo: `http://omnom-worker.dm2e.hu-berlin.de`.
[2]Source code: `https://github.com/DM2E`.

4. RELATED WORK

To enhance the reusability of workflows, their publication as Linked Data has been proposed [2]. DERI Pipes is a simple workflow system for the mashup of Linked Data resources [5]. Many proposals and standards exist for the syntactic and semantic description of web services, for RESTful web services for example ReLL [1]. The combination of RDF and the Linked Data principles to describe RESTful services is also the basis for other systems like PCM-lite [4], Hydra [3], or RESTdesc [8]. A general infrastructure to read and write Linked Data is currently specified by the W3C Linked Data Platform Working Group [6].

5. CONCLUSION

Open workflows are workflows that "live" on the Web: everybody can provide new services, access the provenance chain of all resources and branch existing workflows. We presented a framework which integrates all phases of a typical workflow lifecycle, including the specification of services, their composition to workflows as well as the execution of the workflows. With the specified ontology, all resources in this lifecycle are described. Within this framework, it is possible to track the provenance of all resources and to expose the provenance using W3C PROV.

Acknowledgements: Kai Eckert, Konstantin Baierer, and Dominique Ritze are funded by the European Commission via the project Digitised Manuscripts to Europeana.[3]

6. REFERENCES

[1] R. Alarcon and E. Wilde. Linking data from restful services. In *Proc. of the 3rd Workshop Linked Data on the Web*, 2010.

[2] D. Garijo and Y. Gil. A new approach for publishing workflows: abstractions, standards, and linked data. In *Proc. of the 6th workshop on Workflows in support of large-scale science*, 2011.

[3] M. Lanthaler and C. Gütl. Hydra: A vocabulary for hypermedia-driven web apis. In *Proc. of the 6th Workshop on Linked Data on the Web*, 2013.

[4] L. Panziera and F. De Paoli. A framework for self-descriptive restful services. In *Proc. of the 22nd Int. Conference on World Wide Web companion*, 2013.

[5] D. L. Phuoc, A. Polleres, M. Hauswirth, G. Tummarello, and C. Morbidoni. Rapid semantic web mashup development through semantic web pipes. In *Proc. of the 18th World Wide Web Conference*, 2009.

[6] S. Speicher, J. Arwe, and A. Malhotra. Linked Data Platform 1.0 – W3C Last Call Working Draft 30 July 2013. `http://www.w3.org/TR/2013/WD-ldp-20130730/`, 2013.

[7] V. Stankovski, P. Missier, C. Goble, and I. Taylor. Open workflow infrastructure: a research agenda. In *Proc. of the 1st Int. Workshop on Workflow Approaches to New Data-centric Science*, 2010.

[8] R. Verborgh, T. Steiner, D. Van Deursen, S. Coppens, J. G. Vallés, and R. Van de Walle. Functional descriptions as the bridge between hypermedia APIs and the Semantic Web. In *Proc. of the 3rd Int. Workshop on RESTful Design*, 2012.

[3]`http://dm2e.eu`