

Web Service Composition Reuse through Shared Process Fragment Libraries

David Schumm, Dimitrios Dentsas, Michael Hahn,
Dimka Karastoyanova, Frank Leymann, and Mirko Sonntag

Institute of Architecture of Application Systems, University of Stuttgart,
Universitätsstraße 38, 70569 Stuttgart, Germany
{schumm, karastoyanova, leymann, sonntag}@iaas.uni-stuttgart.de

Abstract. More and more application functionality is provided for use over corporate and public networks. Standardized technology stacks, like Web services, provide abstraction from the internal implementation. Coarse-grained units of Web service composition logic can be made reusable by capturing it as ‘process fragment’. Such fragments can be shared over the Web to simplify and accelerate development of process-based service compositions. In this demonstration, we present a framework consisting of an Eclipse-based process design environment that is integrated with a Web-based process fragment library. The framework enables extracting process fragments, publishing and sharing them on the Web, as well as search, retrieval, and their reuse in a given process. Process fragments can be shared with others using a Web frontend or through a plug-in within the process design environment which is building on Web service technology.

Keywords: Process Reuse, Service Composition, Web Services, BPEL, Process Fragments.

1 Reuse of Web Service Compositions

The main principles in service-oriented applications are providing componentized functionality via stable interfaces abstracting from the implementation details and realizing loose coupling for improved flexibility. Web services are often used as the technology for abstractly describing the service interfaces of components (using WSDL¹) and for composing them in Web service compositions (using BPEL²). Web service compositions allow for the definition of complex orchestration logic and integration of simple and complex application functions that are exposed as Web services. Reusability is one of the main principles of SOA, which however has not been completely realized by existing systems for the purposes of reusing coarse-grained units of service compositions, so-called ‘process fragments’. A *process fragment* is a connected process structure, which can either be modeled from scratch

¹ <http://www.w3.org/TR/wsdl>

² [http://www.oasis-open.org/committees/
tc_home.php?wg_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)

or extracted from a given process. Note that in the field of Web services the terms ‘process’, ‘service composition’, and ‘service orchestration’ are used synonymously. Thus, a process fragment can also be understood as coarse-grained unit of service composition logic. Examples of such fragments have already been presented in [1], where we also discussed the benefits of having multiple process fragment libraries, which can be used to *store* such service composition units and to *share* them between partners in order to ease the creation of service compositions. In this demonstration, we will present a complete and integrated framework, enabling reuse and sharing of Web service compositions over the Web. As platform for process design we rely on a prototype for simulation workflows³ developed in our institute, see also [2]. As platform for fragment sharing we build on the Fragmento library [3] – a repository that provides version management and further functions to support the work with fragments. The integration is mainly achieved through Web service technology, an OSGi-based Eclipse plug-in, and process design environment extensions.

2 Framework Walkthrough

There are four key components in the framework (Figure 1): (i) multiple deployments of the Fragmento Library⁴ and underlying database, which account for versioned process fragment management; (ii) Fragmento’s Web interface, which is not part of the demonstration, allows users to share and reuse process fragments over the Web without a tight integration, using just a common Web client; (iii) the Process Design Environment, which is based on the open source tool Eclipse BPEL Designer⁵; and (iv) an Eclipse plug-in that tightly integrates Fragmento with the design environment. The plug-in connects to the Fragmento library via its Web service interfaces (through generated code skeletons in Java). The plug-in is made available within Eclipse as ‘View’ so that it can be easily positioned between the other graphical components of the design environment.

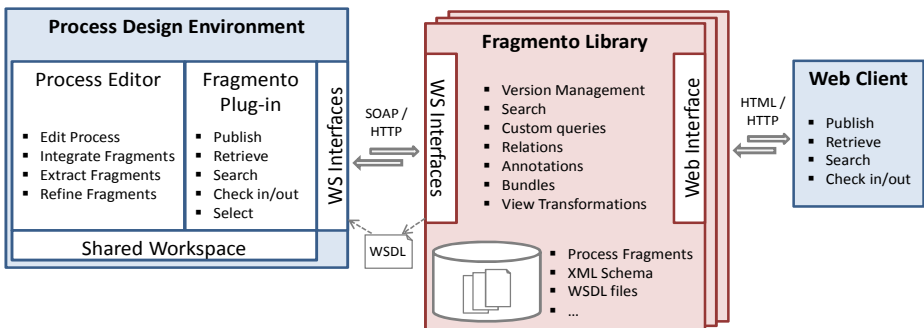


Fig. 1. Overview of the Integration Architecture

³ <http://www.iaas.uni-stuttgart.de/forschung/projects/simtech/index.php>

⁴ <http://www.iaas.uni-stuttgart.de/forschung/projects/fragmento/start.htm>

⁵ <http://eclipse.org/bpel/>

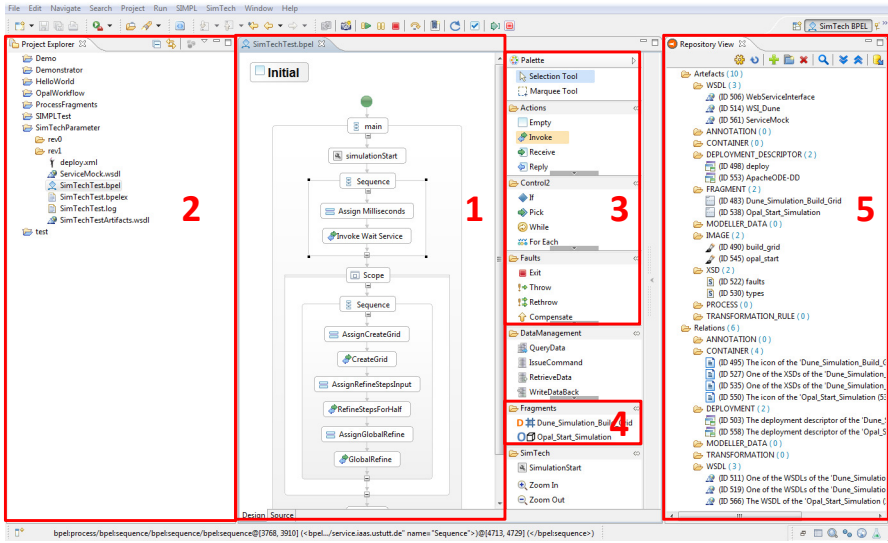


Fig. 2. Main Graphical Components of the Process Design Environment's User Interface

Figure 2 shows the user interface of the process design environment, which is structured into multiple graphical components that can be flexibly arranged:

1. *The editor pane* displays the model of a process or fragment graphically. The user can select process logic and extract and store it locally as fragment. Furthermore, it is possible to insert a fragment into a process by a simple drag-and-drop operation.
2. *The project explorer* shows all files relevant for process design – models of processes and process fragments, deployment descriptors, graphical files (e.g. icons), XML Schema and WSDL interface descriptions. Fragments extracted from a process can be published to a connected fragment library directly from this view.
3. *The editor palette* shows the standard elements that can be used in process design, for instance, activities for service invocation or loop controls.
4. *The fragment palette* shows a list of fragments which are currently selected for usage in process design. The list is refreshed on demand. It is assembled based on contents in a particular directory which is populated by previously exported fragments.
5. *The repository view* reflects the contents of a selected process fragment library in form of a tree control. From this view, processes and process fragments can be published, retrieved, searched, checked in and out, and selected for inclusion in the fragment palette.

3 Demonstration

With the help of examples we will demonstrate the main steps of the approach, namely extraction, publishing, retrieval, selection, and finally integration of process fragments, explained in the following.

Process Fragment Extraction. The editor pane allows the user to select a structured activity, e.g. a BPEL <flow> activity containing several variable assignment and service invocation activities. A right-click on this structure shows a context menu allowing for process fragment extraction. When this function is selected, a connected process fragment is determined, based on the selected activities. A new process file is generated, containing only the fragment and all required variables, XML Schema, and WSDL information of the involved services and information of the process interface related to the fragment. The extracted fragment is opened in a new tab to prepare it for sharing, e.g. placeholders can be added.

Process Fragment Publishing. Publishing of extracted fragments can be directly triggered from within the project explorer by selecting the files generated during fragment extraction. These files can be passed to the Fragmento plug-in by selecting a function in the extended context menu. The plug-in then prepares a set of SOAP messages, each containing a file belonging to the fragment. Based on identifiers returned by Fragmento, interrelations between the artifacts are then created through sending further SOAP messages to Fragmento, i.e. a fragment 'bundle' is defined.

Process Fragment Selection and Retrieval. To be able to select and retrieve fragments using the process design environment or the Web client, a fragment library needs to be chosen, either in the Eclipse plug-in (i.e. using the repository view) by pointing to a WSDL location of a Fragmento deployment or by entering the URL in the Web browser (i.e. using the Web client). In Eclipse the whole contents available in the library are downloaded after connecting to the library. Further work is made on a local copy that is refreshed on demand. Search can be made in Eclipse and in the Web client using keywords, contained text, etc. A fragment can be made available for reuse in the fragment palette through an export function in the repository toolbar. The newest version is chosen by default, but also older versions can be selected. All files of selected fragments are copied to a directory which is shared with the process design environment. Multiple fragments can be added to the palette that way.

Process Fragment Integration. After manual refresh of the fragment palette all exported fragments are shown by their name (and icon if specified). In this palette a fragment can be selected for integration. A click on a location of an opened process defines the place where the fragment should be integrated. During integration, the activities in the fragment are added to that location in the process, the WSDLs and XML Schemas are added to the project, and deployment descriptors are merged.

References

1. Schumm, D., Karastoyanova, D., Kopp, O., Leymann, F., Sonntag, M., Strauch, S.: Process Fragment Libraries for Easier and Faster Development of Process-based Applications. *Journal of Systems Integration* 2(1), 39–55 (2011)
2. Sonntag, M., Karastoyanova, D.: Next Generation Interactive Scientific Experimenting Based On The Workflow Technology. In: *Proceedings of the 21st IASTED International Conference on Modeling and Simulation, MS 2010* (2010)
3. Schumm, D., Karastoyanova, D., Leymann, F., Strauch, S.: Fragmento: Advanced Process Fragment Library. In: *Proceedings of the 19th International Conference on Information Systems Development (ISD 2010)*. Springer (2010)