ICHE 2014, Hamburg - Lehfeldt & Kopmann (eds) - © 2014 Bundesanstalt für Wasserbau ISBN 978-3-939230-32-8

RichWPS Orchestration Environment for Geo Services

F. Bensmann & R. Roosmann
University of Applied Sciences Osnabrück, Osnabrück, Germany

R. Wössner, A. Abecker & C. Heidmann
Disy Informationssysteme GmbH, Karlsruhe, Germany

ABSTRACT: The application of distributed software architectures is a common trend in geospatial analysis. On the basis of service oriented architectures, web services are used to provide and process geospatial data. Especially in the European public administration, the implementation of the INSPIRE directive makes the use of standardized web services mandatory. The web services protocol and interface standards of the Open Geospatial Consortium are well suited for this purpose. This paper describes how web services, initially in the form of OGC Web Processing Services, can be used to efficiently compose and publish complex geodata-processing workflows in an automated and centralized manner. Therefore, the RichWPS orchestration environment and software collection is presented which consists of various components that cooperate to realize workflow modelling and orchestration. The environment is optimized for Web Processing Services, and at the same time provides improved usability for users who are no IT experts. In this paper, the concept and functioning of the service orchestration are described. Following this, the practical application is demonstrated by means of a use case. Finally, a conclusion is drawn to sum up the results and outline both the advantages and drawbacks, closing with a brief outlook for further development.

Keywords: RichWPS, Spatial Data Infrastructure (SDI), Orchestration, Web Service, Web Processing Service (WPS), Workflow

1 INTRODUCTION

With the need for collaborative services increasing and improvements of hard- and software, the application of Service Oriented Architectures (SOA) for geospatial analysis is developing continously. Through using SOA, it is possible to distribute software functionalities to remote servers (Foerster, Schäffer, Baranski, & Brauner, 2011). Thereby, data and processing functionalities can be provided to other users by means of web services (WS). Further, once such a WS is offered it can be used and reused repeatedly, thus providing a framework of sustainability. At the same time, workstations are generally relieved from processing load. Infrastructures that are built in this way to handle geospatial tasks, can be subsumed under the broader concept of Spatial Data Infrastructures (SDI) (Kiehle, 2006). One example for such an SDI is the Marine Data Infrastructure Germany (MDI-DE) which was recently realized, among others by the Federal Waterways Engineering and Research Institute (BAW). Its purpose is the provision of marine geo data to the public.

Furthermore, the INSPIRE directive of the European Union (EU) (EU, 2007) obligates authorities in public administration to participate in the realization of an EU-wide SDI for provision and centralized use of geospatial data.

In this context INSPIRE defines a set of standardized services that are to be implemented. For this purpose the WS of the Open Geospatial Consortium (OGC) are suitable for the geospatial domain (Arbeitskreis Architektur der GDI-DE und Koordinierungsstelle GDI-DE, 2010). The best known INSPIRE-defined services are download services for feature and coverage data. These services can be realized using the Web Feature Service (WFS) or the Web Coverage Service (WCS). Another example are visualization services for which the Web Map Service (WMS) suits well. Although processing services normally play only a minor role, they can be mapped to Invoke Spatial Services and Transformation Services, and can be handled using the Web Processing Service (WPS) (Schut, 2007). WPS is a processing service that was originally designed for geospatial processing but is not necessarily limited to it (Stollberg & Zipf, 2007), (Stollberg & Zipf, 2009). In its 1.0.0 version from 2007, WPS standardizes the interface of the WS. This standardization ranges from the WS operations over the exchanged XML-messages, through to the regulation of usable data types. On one side, this interface is strongly regulated with regard to the overall design while, on the other side, it leaves certain degree of freedom when it comes to what features are actually to be supported.

A WPS provides functionality by means of processes that can be accessed via the operations GetCapabilities, DescribeProcess and Execute. OGC WS (OWS) like the WPS are to a certain degree related to common Web Services of the W3C, although they may differ in significant points. For interoperability reasons WPS are provided with optional extensions for the Simple Object Access Protocol (SOAP) or the Web Service Description Language (WSDL) for its description. All OWS follow common principles, e.g., they share a GetCapabilities operation, from which a description of the properties and abilities of the service can be retrieved. Through their standardized interfaces, that are currently employed various OWS could be harmonized.

The RichWPS orchestration environment presented in this paper focusses on the efficient use of WPS in contexts that conform with the provisions of the INSPIRE directive. Among other things this approach includes providing efficient interoperability with other OWS as well as a user friendly realization. This can be achieved by means of composition of existing services, which is considered a key concept in SOA (Josuttis, 2007) and SDI (Alameh, 2003). A special form of composition is orchestration.

Web service orchestration refers to the coordination of multiple services from a central point of view. Different web services (WS) are executed in a certain order and form a business process (Peltz, 2003). The new business process, or its technical representation the workflow, can again be published as a WS.

Building on this, the RichWPS project realizes an orchestration environment that allows users to model workflows in a user-friendly way and to provide the workflows to other users as WPS process. It can thus be used with a WPS client, e.g., by a geographic information system (GIS). The modelling of workflows takes place graphically in an editor and involves the definition of relations between various WPS processes. That results in the modelling of the data and the control flow between outputs and inputs. Further OWS are to be supported later on.

The need for a user friendly environment is evident. In practice, domain users do not do IT administration since there is a special department for this. Therefore, domain users are not normally allowed or able to deploy new WPS processes, services or workflows on their own, and an IT administrator has to be present for final implementation. This way, however, an efficient development of new workflows including testing, is hardly possible, although this would be crucial for quality assurance and realization of new functionalities. An automated tool that works in a predefined environment and does not affect further IT systems enables the domain user to do his work more independently. With the ModelBuilder, RichWPS aims at providing such a tool.

2 RELATED WORK

In recent years much research effort was invested into the composition of OWS (Stollberg & Zipf, 2007), (Foerster et al., 2011), (de Jesus, Walker, Grant, & Groom, 2012), (Schaeffer, 2009). Since OWS are WS, first composition approaches were done by chaining of WS. Chaining according to ISO19119/Service Architecture standard refers to service chaining as: “A sequence of services where, for each adjacent pair of services, occurrence of the first action is necessary for the occurrence of the second action“ (Schaeffer, 2009). The chaining could be done locally by a client that calls the involved services, or it could be done in a centralized manner through using a workflow or orchestration engine (OE), cf. (Ivanova, 2006). OEs are specialized services that can interpret a workflow model and do the necessary calls. The workflow is then published as an operation by the OE. An important example for a workflow description language is the Business Process Execution Language (BPEL) which can be used together with numerous OEs. Depending on the level of control a user has over the execution of a workflow, distinction is drawn between transparent, translucent and opaque chaining. The application of an OE corresponds to translucent chaining (Alameh, 2003). With a great deal of effort WPS processes have already been chained using BPEL (Kiehle, Heier, & Greve, 2008), (Kiehle, 2006), (Stollberg & Zipf, 2007).

In practice, however, the realization with BPEL is problematic due to technical incompatibilities. For instance, WPS implements remote procedure calls (RPC) different from common W3C WS. Also, the use of SOAP and the application of WSDL distinguish the WPS from common W3C WS (Kiehle et al., 2008). Thus, the use of BPEL is an option, but it requires numerous adaptions (Kiehle, 2006). Also, BPEL relies on WSDL as a core standard. However, the provision of a WSDL description for a WPS is not mandatory (Brauner, Foerster, Schaeffer, & Baranski, 2009), (Stollberg & Zipf, 2007). For this reason, the application in different open source implementations of WPS is solved in different ways. The picture is similar for SOAP. SOAP allows for the use of further WS standards, the so called WS-* standards. However, where in WPS their application is optional, in practice this means that many variations of WPS servers with different properties and capabilities have to be considered. So orchestrating WPS with a W3C WS conform OE is a rather complicated task.

On a more conceptual level, chaining also means that no control structures, e.g., for conditional execution or loops or error handling, can be used although these are essential for scientific application areas (Akram, Meredith, & Allan, 2006).

Some approaches realize orchestration by using OGC driven interoperability proposals. One approach frequently used is the covering of an OE behind a WPS which serves as a wrapper or a façade (Zhao, Di, & Yu, 2012), (Schaeffer, 2009). The WPS provides the workflows of the BPEL OE as WPS processes to the outside. Calls to these WPS processes are then redirected to the respective operation of the OE without the knowing of the client. This approach has the advantage, that from the view of the SDI, every geospatial processing functionality can be accessed in a standardized way (Stollberg & Zipf, 2009). So, a workflow is an INSPIRE-conform WPS process. In this context, a deploy and undeploy mechanism for WPS processes known as WPS-T has been elaborated and tested (Schaeffer, 2008). It was used to deploy BPEL scripts onto the covered OE. WPS-T is considered to be part of the upcoming WPS 2.0 standard.

Further known approaches are delivered with the WPS 1.0.0 standard itself. One of these is the use of nested executes as described in the WPS standard (Schut, 2007), (OpenPlans, 2014). In this approach the chaining information about subsequent processes is transmitted with the request to every process. This could result either in a centralized chaining or a cascading chaining (Stollberg & Zipf, 2007). However this approach lacks a centralized provision of the workflow. Another option is to embed the required functionality, meaning the calls to external services, as a compilation solution into a process (Schut, 2007), (Stollberg & Zipf, 2007). Yet this is a rigid approach that requires experienced informaticians and relies on compatible clients.

Apart from orchestration, and thinking of modelling and usability, various projects have come up with modelling tools to support users. In general, most of them are tightly coupled to BPEL and to the W3C WS standard solutions they apply. Still modelling is rather complex and expert knowledge in informatics is required (de Jesus et al., 2012), thus inhibiting broader uptake by domain users who are not IT experts. One more elaborated approach that makes use of the Taverna workbench is described by de Jesus (de Jesus et al., 2012). This work has been an example for the ModelBuilder presented in this paper.

Taken together, much work has already been conducted to orchestrate WPS and other OWS. Despite these efforts all approaches have in common that they intent to apply traditional standard solutions borrowed from the W3C domain. For various reasons this results in complex adaptions. Also, the techniques applied are far from being suited for efficient use by domain experts because they generally require technical knowledge from a different subject. Some works, however achieve good results by focusing on OGC driven approaches but again these are regularly lacking valued functionality from traditional approaches.

3 THE RICHWPS APPROACH

RichWPS' approach aims at the harmonization of WPS and other OWS involved in orchestration and, at the same time, intents to avoid the afore-mentioned shortcomings. The basis for this is its focus on a logical subset of OWS, and the exploitation of their standardized behavior and self-description property. Put in concrete terms, this means that, through applying an optimized and tailored orchestration language a workflow description can be created that works with a minimum of information on involved OWS. This language, the RichWPS Orchestration Language (ROLA) is the key concept of this approach. Around this language, a sophisticated software suite is developed that is presented in the remainder of this chapter.

The orchestration environment consists of three components: Firstly, a specialized OE that is placed within a WPS server and is able to execute ROLA described workflows. This component is called RichWPS server. Secondly, the environment provides the ModelBuilder, a graphical editor for modelling, deployment and testing of workflows. Thirdly there is a directory server, called the SemanticProxy, which can be used for searching for information about available services and WPS processes during modelling.

The following illustration shows an overview of the involved components. Arrows indicate a use-relation. Alongside the afore-mentioned components, there are the business services, i.e. the services from which a workflow is composed. Finally, the illustration includes an exemplary process user as well.

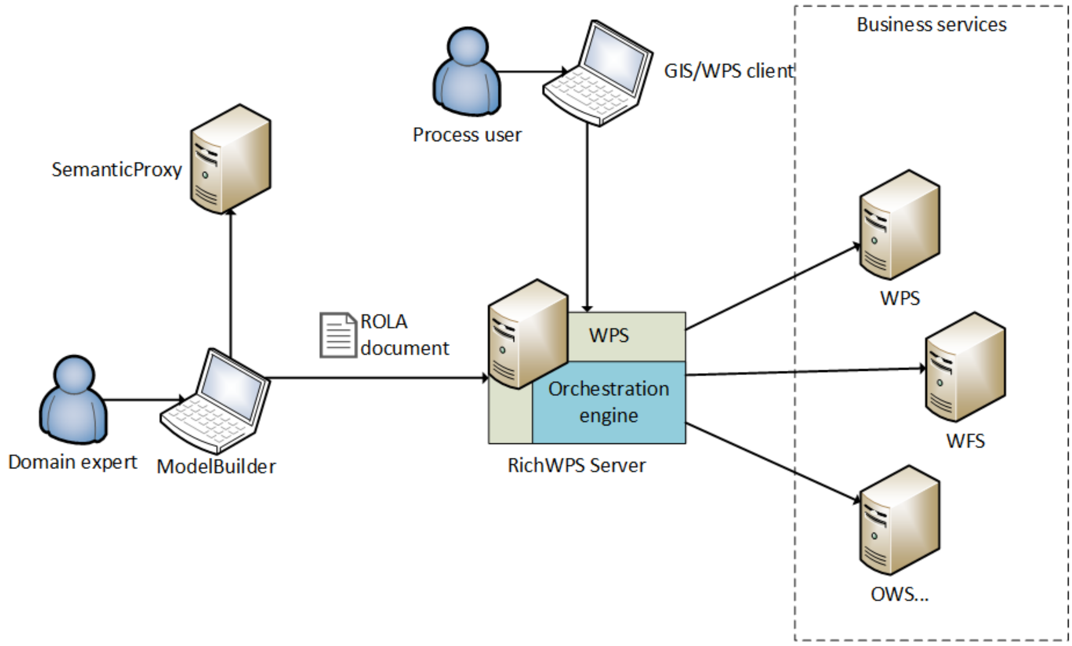

Figure 1. RichWPS component setup in application with business services.

3.1 *Modelling*

In a typical work sequence the domain expert uses the ModelBuilder to design workflows. For this he connects to the SemanticProxy to receive information on available processes in the target network. Once this information is gathered the expert can model a workflow. Figure 2 shows a screenshot of the Model-Builder. Available model elements (processes, workflow inputs and workflow outputs) are shown on the left, while the editor pane into which the elements can be drawn via Drag and Drop is located in the center. Detailed information about selected processes and services is shown on the right.

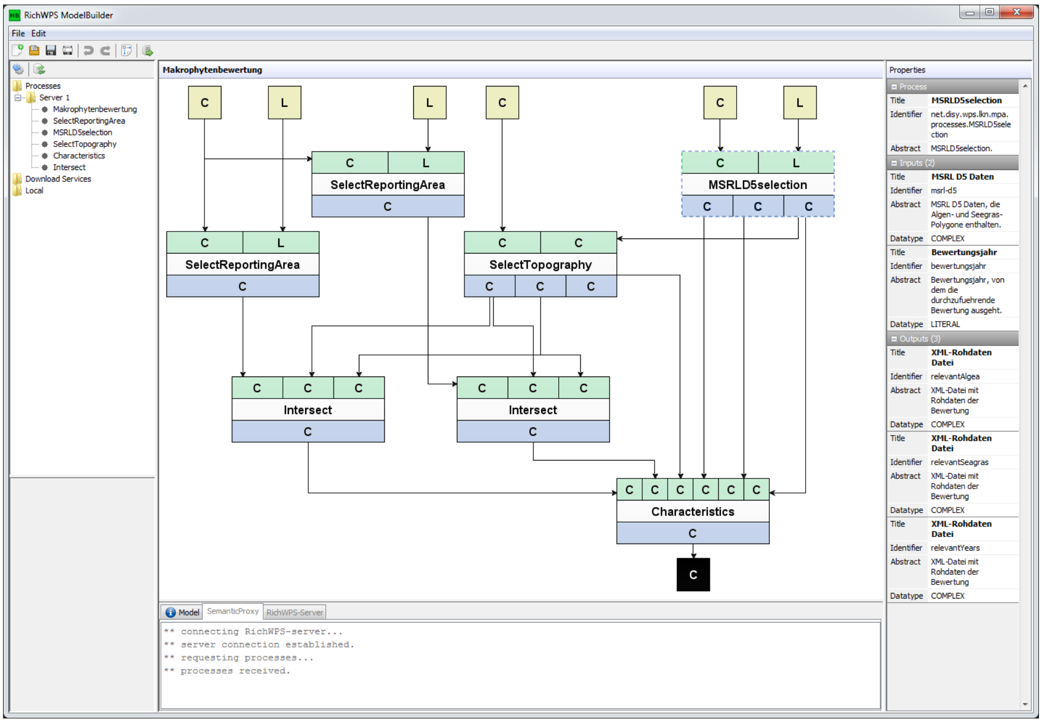

Figure 2. Screenshot of the ModelBuilder.

The Bottom area is intended for outputs of the deployment and testing.

During modelling outputs are connected to inputs in order to describe the data flow. The ports accept the three basic data types defined in WPS 1.0.0: Literals, BoundingBox and Complex. Complex data types can encapsulate any arbitrary data type.

When a workflow has been designed, the graphical model is translated into a ROLA script and transmitted to the server. Upon receipt, the server can then publish the workflow as a WPS process.

3.2 *Orchestration*

For the execution of workflows the client- and server side required components are implemented, including the OE.

The RichWPS server, is based on a 52°North WPS server[1] that is extended with an OE, able to interpret ROLA scripts. The scripts can be deployed using the WPS-Transactional (WPS-T) interface. This way, a mechanism that can be reused for other processes is obtained. WPS-T allows for processes to be removed from the server.

ROLA enforces a strict separation between the workflow description and the process signature that is to be used for publishing. This makes it suitable for WPS-T and increases readability of the script. Like other common programming languages ROLA works with a list of commands that has to be processed sequentially. At creation time the order of the commands is determined by the model diagram. Up to this point in time, workflows can be described using the following elements: References, assignments, bindings and execution statements. Listing 1 shows a simple example of a ROLA script.

```
bind process net.disy.wps.lkn.processes.SelectReportingArea to lkn/selreporting
var.area = "NF"
execute lkn/selreporting
with
  var.area as area
  in.reportingareas as reportingareas
store
  reportingarea as out.reportingAreasNF
```

Listing 1. Example of a ROLA script.

When selecting the language elements particular attention was paid that they are close to natural language, though the most common use case should be the using ROLA indirectly by means of the ModelBuilder. At the moment, the ROLA is still subject to various modification influences. On the one hand, it shall be extended with further elements to steer the flow of control; on the other hand it is expected that the upcoming WPS 2.0 standards will cause numerous extensions. At this time, the support of further OWS and W3C WS is not even taken into account. ROLA is considered to be a pure orchestration language. Verification whether two processes are logically compatible or at least whether the exchanged data types are compatible is not realized. For this kind of tasks, the ModelBuilder was developed. Therefore a test mechanism was designed to be able to deploy a ROLA script on-the-fly on the server, start a test run, get the results and undeploy the workflow.

When calling a process that represents a workflow the OE is started and provided with the respective ROLA script. At the beginning, the OE receives the process' arguments and stores them in the current execution context for later use at calls. It then processes the commands of the ROLA script in a step by step fashion, whereas calls to remote processes constitute the actual orchestration. Produced intermediate results are stored in the current execution context just like the arguments. After finishing the last process the OE returns the result to the user and clears the execution contexts. In case of an error, due to incorrect process use, network error etc. the OE stops the execution and returns a WPS exception.

3.3 *Example processing of use case*

A use case has already been implemented and could successfully prove that the environment can handle realistic use case scenarios satisfactorily. The treated scenario belongs into the context of the tasks needed for the realization of the Marine Strategy Framework Directive (MSFD) and the Water Framework Directive (WFD) issued by Schleswig-Holstein's Government-Owned Company for Coastal Protection, National Parks and Ocean Protection (LKN). Together with further North Sea resident countries the state of

[1]http://52north.org/communities/geoprocessing/wps/

the North Sea is recorded, evaluated and reported on regularly. For reasons of transparency, traceability and comparability this shall be realized using WPS.

The scenario deals with the assessment of macrophytes in the Wadden Sea of Schleswig-Holstein. The spreading of sea weed and algae is a quality factor involved in assessing the eutrophication state for the WFD. Reports contain various statements about particular characteristics of the Wadden Sea. The procedure is described in more detail in (Dolch, Buschbaum, & Reise, 2009) but in order to provide an impression of the task, the applied data-processing steps are outlined here.

- Statements are made for a defined reference space.
- Statements about part parameters can refer to different, overlapping spaces; they are mapped to each other using a geometrical intersection.
- Statements about spaces are made binary (true/false), quantitatively or gradually – often in relation to the area.
- Exceedances or shortfalls of thresholds mark changes in quality.
- Often multiple factors are incorporated in the calculation. The factors need to be harmonized and mapped to a common scale.
- A classification is made for the determination of assessment levels.

The use case has already been implemented exemplarily as tightly integrated WPS process for the 52°North WPS server (Wössner, 2013). For the test the process is subdivided into several smaller processes that are deployed on a WPS server. Using RichWPS the processes will be recomposed into a workflow. Since new interfaces with new data types originated at the processes, additional parsers and generators have to be implemented. At this point the plugin mechanism of the 52° WPS server comes in handy.

The resulting processes are: SelectReportingArea, MSRLD5Selection, SelectTopography, Intersect and Characteristics

The processes are recomposed into an equivalent workflow as shown by the following diagram. SelectReportingArea and Intersect is used multiple times therein.

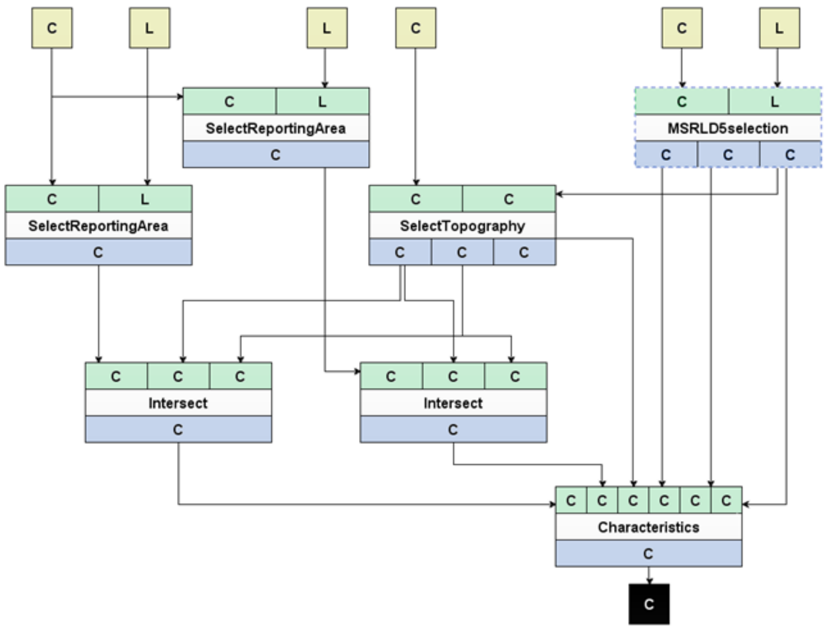

Figure 3. Workflow diagram for the macrophyte assessment scenario.

As mentioned before, the diagram is converted into a sequential schedule description during the translation of the model into ROLA. Thereby the execution order of the processes is determined, which could look like illustrated in Figure 4. The illustration shows the WPS servers with their respective processes and the RichWPS server with the workflow. The arrows indicate the calls of the RichWPS server to the other WPS servers whereby the numbers indicate the order of the calls. While, in theory calls 1-3 and 5-6 could be executed in parallel, this feature will be implemented later on.

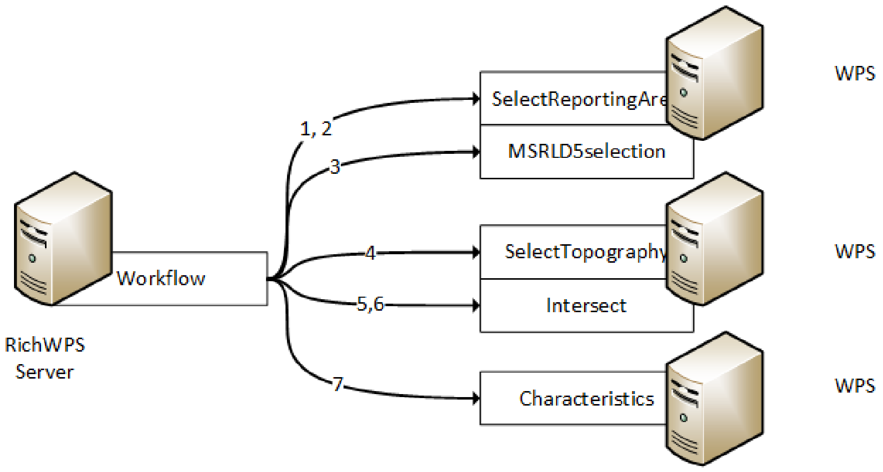

Figure 4. Execution order.

The workflow is published as a WPS process. A test confirmed that the execution is actually possible.

4 SUMMARY AND OUTLOOK

This paper presents the RichWPS orchestration environment. Using an actual use case, it is shown what the application could look like in practice. The RichWPS orchestration environment starts in SDIs with the WPS and is to be extended to include further OWS, and later on, W3C WS. Three vital components are cooperating to achieve the orchestration: An OE, a model editor for the design of workflows, and a directory service for finding available services.

If compared to established solutions, it is possible to gain a significant reduction of workflow description complexity by considering the specialties of WPS. First and foremost, this advantage benefits the user who models the workflow. Consequently RichWPS enables users to define, deploy and test workflows without the assistance of IT experts.

With regard to practical application, a use case implementation revealed, that a coherent set of basis WPS processes and data types is needed. Otherwise a new process respectively a new parser- generator pair has to be implemented for every task and data type. Currently, there are efforts inside the OGC to identify required data types for different domains. Also a basic set of processes is often provided along with the WPS server, in how far they can be used in a specific domain depends on a variety of factors and has to be evaluated on a case-by-case basis.

It was shown that the overall concept is feasible. However, a few extra functions would add value for practical application. At the moment, extensions for debugging and performance profiling are considered as a supplement to testing. Additionally a monitor for the measuring of WPS performance characteristics is currently under development. With this, the quality of service of WPS processes can be regarded at composition time. Or it can be used to realize a semi- or fully automatic optimization of workflows by replacing individual process with processes that have a better performance.

How the orchestration environment is actually going to evolve will depend on validations with more use cases.

REFERENCES

Akram, A., Meredith, D., & Allan, R. (2006). Evaluation of BPEL to Scientific Workflows. In *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)* (Vol. 1, pp. 269–274). IEEE. doi:10.1109/CCGRID.2006.44

Alameh, N. (2003). Chaining geographic information web services. *IEEE Internet Computing*, *7*(5), 22–29. doi:10.1109/MIC.2003.1232514

Arbeitskreis Architektur der GDI-DE und Koordinierungsstelle GDI-DE. (2010). Architektur der Geodateninfrastruktur Deutschland. Retrieved from http://www.geodaten.niedersachsen.de/download/63655/Architekturkonzept_der_Geodateninfrastruktur_Deutschland_-_Version_2.0_vom_09.09.2010.pdf

Brauner, J., Foerster, T., Schaeffer, B., & Baranski, B. (2009). Towards a research agenda for geoprocessing services. In *12th AGILE International Conference on Geographic Information Science (2009)* (Vol. 1, pp. 1–12). Retrieved from http://www.ikg.uni-hannover.de/fileadmin/agile/paper/124.pdf

De Jesus, J., Walker, P., Grant, M., & Groom, S. (2012). WPS orchestration using the Taverna workbench: The eScience approach. *Computers & Geosciences*, *47*, 75–86. doi:10.1016/j.cageo.2011.11.011

Dolch, T., Buschbaum, C., & Reise, K. (2009). Seegras-Monitoring im Schleswig-Holsteinischen Wattenmeer 2008. ... *Für Das Landesamt Für Natur Und* …. Retrieved from http://www.blmp-online.de/Monitoringhandbuch/Dokumente/Makrophyten/Dolch_Reise_2008-LANU_SeegrasendberichtZu2007.pdf

EU. (2007). Directive 2007/2/EC of the European Parliament and of the council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). *Official Journal of the European Union*, *50*, 1–14. Retrieved from http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF

Foerster, T., Schäffer, B., Baranski, B., & Brauner, J. (2011). Geospatial Web Services for Distributed Processing - Applications and Scenarios. In P. Zhao & L. Di (Eds.), *Geospatial Web Services: Advances in Information Interoperability* (pp. 245–286). doi:10.4018/978-1-60960-192-8.ch011

Ivanova, E. (2006). Orchestrating Web Services–Standards and Solutions. ... *" Mathematics, Informatics and Computer Sciences"-St.* …. Retrieved from http://sai.infotel.bg/PROJECTS/VispBG/files/publications/V.Tarnovo06_9.pdf

Josuttis, N. (2007). *SOA in practice: the art of distributed system design*. (S. St. Laurent, Ed.). O'Reilly Media, Inc.

Kiehle, C. (2006). Business logic for geoprocessing of distributed geodata. *Computers & Geosciences*, *32*(10), 1746–1757. doi:10.1016/j.cageo.2006.04.002

Kiehle, C., Heier, C., & Greve, K. (2008). Requirements for Next Generation Spatial Data Infrastructures-Standardized Web Based Geoprocessing and Web Service Orchestration. *Transactions in GIS , 2007*, *11*(6), 819–834.

OpenPlans. (2014). WPS Processes - GeoServer 2.5.x User Manual. Retrieved June 24, 2014, from http://docs.geoserver.org/stable/en/user/extensions/wps/processes.html#process-chaining

Peltz, C. (2003). Web services orchestration and choreography. *Computer*, *36*(10), 46–52. doi:10.1109/MC.2003.1236471

Schaeffer, B. (2008). Towards a transactional web processing service. *Proceedings of the GI-Days, Münster*. Retrieved from http://www.gi-tage.de/archive/2008/downloads/acceptedPapers/Papers/Schaeffer.pdf

Schaeffer, B. (2009). *OGC OWS-6 Geoprocessing Workflow Architecture Engineering Report*. *OGC Public Engineering Report*. Retrieved from https://portal.opengeospatial.org/files/?artifact_id=34968

Schut, P. Web Processing Service V 1.0.0 (2007). Open Geospatial Consortium. Retrieved from http://portal.opengeospatial.org/files/?artifact_id=24151

Stollberg, B., & Zipf, A. (2007). OGC Web Processing Service Interface for Web Service Orchestration Aggregating Geo-processing Services in a Bomb Threat Scenario. *Web and Wireless Geographical Information Systems*, 239–251. doi:10.1007/978-3-540-76925-5_18

Stollberg, B., & Zipf, A. (2009). Development of a WPS Process Chaining Tool and Application in a Disaster Management Use Case for Urban Areas. *Urban and Regional Data Management: UDMS 2009 Annual*, 269. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Development+of+a+WPS+Process+Chaining+Tool+and+Application+in+a+Disaster+Management+Use+Case+for+Urban+Areas#0

Wössner, R. (2013). *Untersuchungen zur praktischen Nutzbarkeit des OGC Web Processing Service (WPS) Standards*. Hochschule Karlsruhe Technik und Wirtschaft.

Zhao, P., Di, L., & Yu, G. (2012). Building asynchronous geospatial processing workflows with web services. *Computers & Geosciences*, *39*, 34–41. doi:10.1016/j.cageo.2011.06.006