

A GEOSPATIAL WEB SERVICES COMPOSITION FRAMEWORK SUPPORTING REAL-TIME STATUS MONITORING

Lan You^{a,b}, Zhipeng Gui^{a,c}, Wei Guo^a, Shengyu Shen^a, Huayi Wu^{a,*}

^a LIESMARS, Wuhan University, 129 Luoyu Road, Wuhan, 430079, China

^b School of Mathematics and Computer Science, Hubei University, 11 Xueyuan Road, Wuhan, 430062, China

^c Center of Intelligent Spatial Computing for Water/Energy Science, 4400 University Dr., Fairfax, 22030, VA, USA

Commission IV, WG IV/5

KEY WORDS: Geospatial Web Services Composition, Web Service Agents, Real-Time Monitoring

ABSTRACT:

Geospatial web services composition becomes one of the main solutions for complex computing in the GIS realm with the development of information interoperability and advanced IT technologies. Standard geospatial web services only have two simple statuses: success or failure. However the procedures for geospatial information processing and analysis always feature intensive data, complex computation, and long processing times. Thus, standard geospatial web services composition only provides basic functions and cannot fully satisfy geospatial information processing and analysis needs. The problems with standard geospatial web services are numerous, including difficulties in controlling and monitoring geospatial web services, composition optimization, and error tracking. The execution status of geospatial web services composition must be monitored becoming the basis for further optimization of the geospatial web services composition model. This paper proposes a framework for geospatial web service composition that supports real-time status monitoring. This framework effectively integrates geospatial web services with status interfaces and dynamic monitoring technology. Experiments show that this framework can effectively monitor execution information, reminding users of bottleneck information thus providing the foundation for further improvements in the model's execution efficiency.

1. INTRODUCTION

One of the developing trends in web service technology is web services composition. Various web services composition specifications have been proposed, e.g., WS-BPEL (Web Service - Business Process Execution Language, OASIS, 2007), WS-CDL (Web Services Choreography Description Language, W3C, 2004a), and WSFL (Web Services Flow Language, IBM, 2001), that have made standardized web service procedures more and more common. As IT technologies matured, web services composition gradually diffused into the GIS realm and is becoming an important solution for geospatial information processing and analysis. A characteristic of standard web services is that they have a single processing status. There are only two execution statuses: success or failure in an opaque processing procedure. For geospatial information processing and analysis, which typically includes intensive data processing, complex computation, and long processing times, a standard geospatial web services composition can only support basic functions. The problems with standard geospatial web services are numerous, including difficulties in controlling and monitoring geospatial web services, composition optimization, and error tracking. For example, when one standard geospatial web service composition executes unusually slowly, no hint or information is provided, making it difficult to find the exact solution to optimize the geospatial web services composition to speed up the execution procedure.

Therefore, how to capture the exact execution status of geospatial web services composition and how to monitor and

visualize the running status are two main problems that must be urgently solved.

Some research on geospatial web services composition has been conducted. Nadine (2003) and Friis-Christensen (2009) compared and analysed platform architecture models for geospatial web services composition. Semantic reasoning and AI planning based automatic/semi-automatic modelling has become a hot topic for research with the development of the semantic web and ontology theory. Lemmens (2007) investigated how to establish low-level semantic descriptions for automatic modelling of geospatial web services composition. Others (Di, 2006; Yue, 2007; Chen, 2009) proposed ontology-driven modelling schemes, in which OWL-S (W3C, 2004b) based abstract models are created semi-automatically, and transformed into WS-BPEL for execution. SWING (Andrei, 2008) is a semantic based modelling framework, that covers the entire lifecycle of geospatial web services composition. SAW-GEO (Gobe, 2007) is a prototype system, which combines a visual modelling solution based on industry specifications and semantic reasoning. Li (2010), Wu (2010), and Yang (2011) have done some research on optimizing the integrated framework for supporting geospatial web services.

Most of this research work focuses on the model design and invocation of geospatial web services composition. This work places emphasis on the simplification of the modelling procedure and invoking models, but ignores monitoring running status. During the execution procedure for geospatial web service composition, users cannot acquire the execution status

* Corresponding author. Email: wuhuayi@whu.edu.cn, Tel: +86 27 68778311, Fax: +86 27 68778969

or monitor the bottleneck information, making it difficult to optimize geospatial web services composition models. Moreover, when the geospatial web services composition model execution fails, users cannot track errors, resulting in an unpleasant user experience. These deficiencies show that most of the current geospatial web service composition frameworks just work for building and execution of the geospatial web service model, but contribute little to the optimization and reuse of models.

Given these deficiencies, a framework for geospatial web service needs a composition supporting real-time status monitoring. It is necessary to establish a platform that integrates geospatial web services composition modelling, deployment, execution and monitoring.

This paper proposes a framework for geospatial web service composition supporting real-time status monitoring. This framework effectively integrates the web services with status interfaces and dynamic monitoring technology. The framework consists of four parts: client, engine for geospatial web service composition, geospatial web services agent, and geospatial web services with status. It has the following features:

- Provides a visual module in the client, through which the geospatial web service composition running status can be monitored in real-time in order to track the component services and optimize the composition intuitively.
- Implements geospatial web services with interfaces through which execution details can be acquired such as percentage of a task completed, the number of subtasks, and each subtask's execution details, etc.
- A geospatial web services agent was designed to be in charge of collecting the running status of component geospatial web services.
- Modeling, deployment, monitoring, and invocation functions are integrated for geospatial web services composition.

The remainder of this paper is organized in three sections. Section 2 introduces the design of the framework for geospatial web service composition supporting real-time status monitoring; the components and their functions are presented in detail. Section 3 gives an example the framework in use. Section 4 is a brief conclusion and future work.

2. FRAMEWORK DESIGN

The design for the framework for geospatial web service composition supporting real-time status monitoring is as shown in Figure 1.

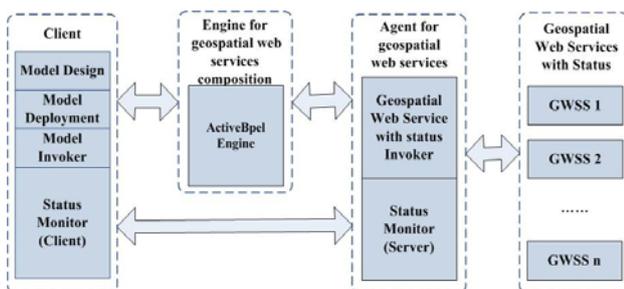


Figure 1. Framework Design

2.1 Client

The main functions of this part involve the design, deployment, and invoking of the geospatial web service composition, as well as communication with web services agents in real time. The client part integrates the status monitor that visualizes the geospatial web services' running progress. These are the components for geospatial web services composition.

2.1.1 Model Design: Constructs a model for geospatial web services composition by selecting some geospatial web services and building links with the WS-BPEL rule. WS-BPEL is one kind of web service composition standard that provides a notation for describing web service interactions as business processes.

2.1.2 Model Deployment: Provides a function for a geospatial web services composition model which can be deployed to the workflow engine and supports WS-BPEL.

2.1.3 Model Invoker: Invokes the geospatial web services composition model previously deployed in the workflow engine and receives the response.

2.1.4 Status monitor (Client): The status monitor in the client updates the status bars, each of which represents a geospatial web service, when the model invoker has invoked the model of geospatial web services composition successfully. Dynamic execution status information is shown in real time to monitor the execution procedure of the model even if the running procedure fails. The status monitor in the client acquires the updated status by communicating with the status monitor in the geospatial web services agent.

2.2 Engine for geospatial web services composition

This framework adopts the ActiveBpel Engine 5.0.2 as the engine for geospatial web services composition. The ActiveBpel Engine is in charge of publishing and the execution of composition models. ActiveBpel Engine is an open source workflow engine.

2.3 Agent for geospatial web services

This part handles invoking and responding messages between the geospatial web service composition engine and the component services, as well as cycle queries the status of component web services and caches for the client invoking. The agent consists of two main components: a geospatial web service invoker and a status monitor. The geospatial web service invoker manages the invocation of the geospatial web service with status and communicates with the status monitor in the server side, while the status monitor in the agent cycle queries the geospatial web service with status and caches the status for the status monitor in the client.

2.4 Geospatial Web Services with Status (GWSS)

In this framework, geospatial web services must provide status query interfaces, through which web services agent can query the running status in real time.

Those interfaces are used for requesting status data from the geospatial web services. The status monitor in the agent server is developed on the basis of them. Those interfaces include *getruninfo*, *getmachineinfo*, etc. *Getruninfo* interface is used to

get the running status and *getmachineinfo* interface is used to get the machine's information that is running the geospatial web service.

3. EXPERIMENT

A prototype for the geospatial web service composition framework supporting real-time monitoring was developed. The client of the experiment system was developed on Eclipse 6.0, and was implemented with Java RCP techniques. The prototype took ActiveBpel engine for the geospatial web services composition. A java servlet was implemented as the geospatial web services agent. Tomcat 5.5 was used as the service container. The geospatial web services with status interfaces are implemented with C++ program language.

There are three geospatial web services with statuses deployed on a local-area network (LAN). They include an image filter, a SVM (Support Vector Machine) –based image classification and a statistical service. The ActiveBpel Engine is deployed on one server. In order to monitor geospatial web services composition, an agent for geospatial web services is deployed on the LAN. A geospatial web services composition model is designed including feature classification and statistical analysis using the three geospatial web services on the client. The raw dataset input is an image, and the output are the statistical results with the given features classification. In this experiment as the study area, we use image data covering the Poyang lake area, which is about 246MB. Figure 2 shows the model example diagram.

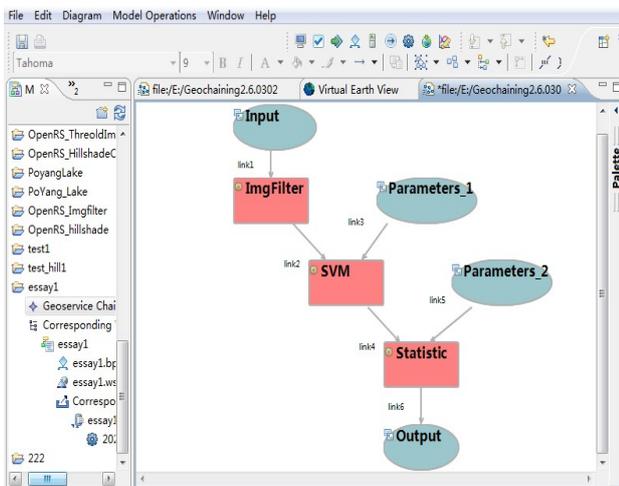


Figure 2. Geospatial Web Services Composition Model

When the client model invoker invokes the model, the client status monitor will show the status such as the progress of each component service, the count of subtasks, and status of each subtask. The detailed execution status of the model is described in Figure 3.

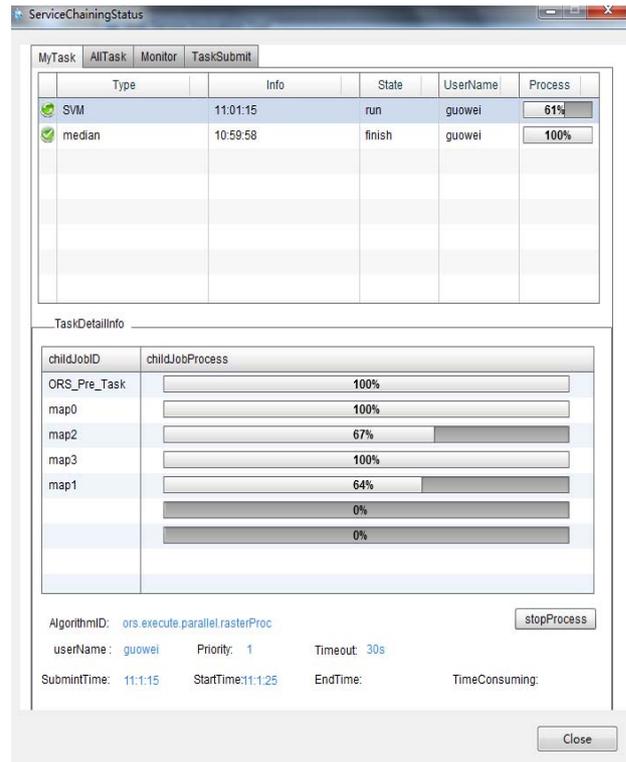


Figure 3. Status Monitor in the Client

In the top frame of Fig.3, there are only two geospatial web services displayed, because statistics service is the last service in the geospatial web service composition. For this service, the input data is the output of the prior two services. In figure 3, it can be seen that the SVM service has finished the 61 percent of the task. The SVM service's task has been divided into four subtasks as shown in the bottom frame of the Figure 3 including map0, map1, map2, map3. In Figure 3, it can be seen that, among the four subtasks, map0 and map3 have finished; map1 has completed 64 percent of the whole subtask; and map2 has finished 67 percent of the whole subtask. As illustrated in Figure 3, the Image filter service has finished, but the statistical service has not begun. On the backend, the status monitor in the agent server works together with the status monitor module in the client.

Figure 4 shows the server information where the geospatial web services are running, when the geospatial web services composition is running. In Figure 4, the frame of *Machineinfo* includes machine name, the count of the processes running on the server, the CPU usage information, and the memory usage information. The *Processinfo* frame displays the name of the running geospatial web service, the job's status and other server information.

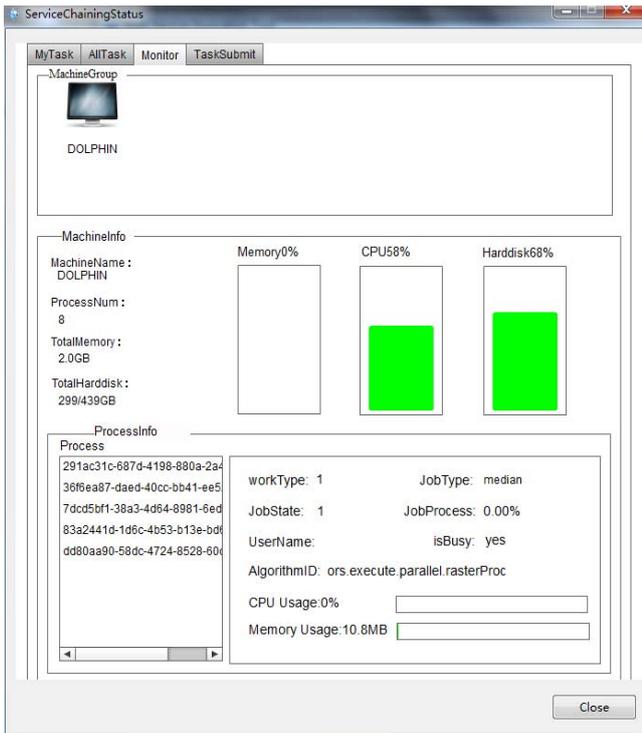


Figure 4. Server information

Figure 5 shows the final information when the geospatial web services composition has completed. It can be seen that there are three geospatial web services displayed in the top frame. The *state* column of each geospatial web service displays *finish*, which means that the service has completed its task. It also can be seen in Figure 5 that all the subtasks (*ORS_Pre_Task*, *simple task*, *ORS_POST_Task*) have been completed.

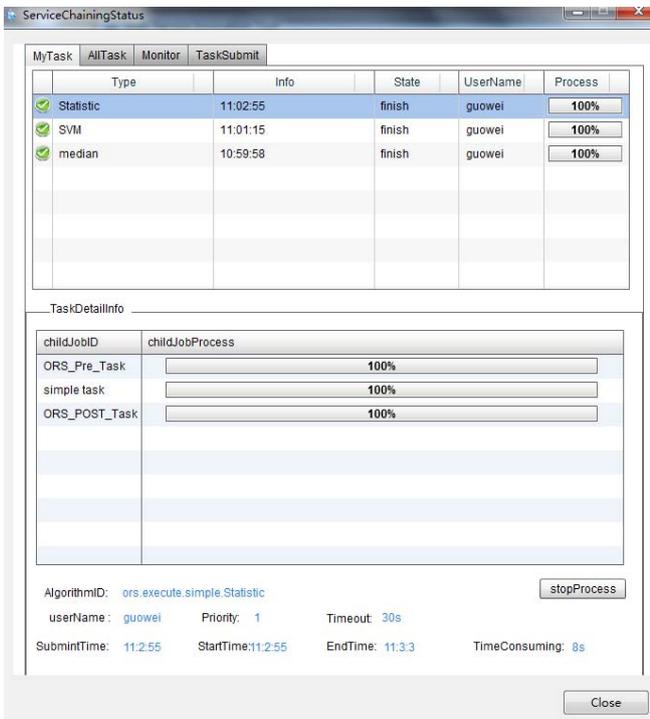


Figure 5. Final status information

The whole procedure for the geospatial web services composition costs about 3 minutes. The SVM service occupied 2 minutes; more than half of the whole procedure. The status monitor shows that the SVM service costs too much time, so a solution must reduce its processing time in order to improve the model's execution efficiency. If the framework does not support the status monitor in real-time, users do not know what to do or how much of the geospatial web services composition model has been executed. Thus, there will be a poor user experience.

Experiments show that our framework not only builds and invokes the geospatial web services model, but also satisfies the need to monitor execution status in real time. In our framework, users can acquire bottleneck information during the model execution procedure, which is reflected back into the design of models to achieve targeted optimization. This promotes consistent optimization of the models and improves user experience.

4. CONCLUSION AND FUTURE WORK

In the GIS realm, geospatial information processing and analysis always have features including intensive data, complex computation, and long processing time. The blooming of IT technologies has made web service composition one of the main solutions to the problems of complex computing. However, standard geospatial web services composition can only provide basic functions. Problems such as the difficulties in controlling and monitoring geospatial web services, and composition optimization must be solved. A geospatial web service composition framework supporting real time monitoring is a solution to some of these problems.

Most of the recent research focuses on the simplification and automation for building geospatial web services composition models, and ignores procedures to optimize models. A framework for geospatial web services composition supporting real-time status monitoring can not only build, deploy, and execute geospatial web service composition but also monitor the execution status. The framework consists of four parts: a client, an engine for geospatial web service composition, a geospatial web services agent, and geospatial web services with status interfaces. Among these parts, the status monitor module is the important part both on the client and on the agent server. On one side, the module on the client communicates with the module on the agent server and shows the progress dynamically; on the other side, the module on the agent server cycle queries the status interfaces provided by geospatial web services delivering the status for collecting the real-time status.

Experiments demonstrate that the framework can really show execution information, reminds users with bottleneck information and provides a foundation for further improvements in the efficiency of the model's execution.

In the future, standard geospatial web services will be researched, so more interoperable OGC standardized services can be implemented. The goal will be to construct a fused geospatial web services composition including not only the web services with status interfaces but also those with no status interfaces. That will make the framework more expansive.

5. REFERENCES

- Andrei, M. et al., 2008. SWING: An Integrated Environment for Geospatial Semantic Web Services. In: *ESWC 2008*. LNCS 5021, S. Bechhofer, Ed., ed.: Springer: Berlin, pp. 767-771.
- Chen, J., et al., 2009. Use of grid computing for modelling virtual geospatial products. *International Journal of Geographical Information Science*, 23(5), pp. 581-604.
- Di, L., et al., 2006. Ontology-driven automatic geospatial-processing modelling based on web-service chaining. In: *Proceedings of the Sixth Annual NASA Earth Science Technology Conference*, 7pp.
- Friis-Christensen, A., et al., 2009. Service chaining architectures for applications implementing distributed geographic information processing. *International Journal of Geographical Information Science*, 23(5), pp. 561-580.
- Gobe, H., et al., 2007. Semantically-assisted geospatial workflow design. In: *Proceedings of the 15th international symposium on Advances in geographic information systems*, Seattle, Washington, 7pp.
- IBM, 2001. "Web Services Flow Language (WSFL 1.0)". <http://www.itee.uq.edu.au/~infs7201/Assessments/Assignment%201%20Material/WSFL.pdf>.
- Lemmens, R. et al., 2007. Enhancing Geo-Service Chaining through Deep Service Descriptions. *Transactions in GIS*, 11(6), pp. 849-871.
- Li, Z., Yang, C., et al., 2010. An optimized framework for seamlessly integrating OGC web services to support geospatial sciences, *International Journal of Geographic Information Science*, 25(4), pp. 595-613.
- Nadine, A., 2003. Chaining geographic information web services. *IEEE Internet Computing*, 7(5), pp. 22-29.
- OASIS, 2007. "Web Services Business Process Execution Language Version 2.0 (WS-BPEL 2.0)". <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- W3C, 2004a. "Web Services Choreography Description Language Version 1.0". In: N. Kavantzaz, D. Burdett, G. Ritzinger. (Eds.) W3C. <http://www.w3.org/TR/ws-cdl-10/>.
- W3C, 2004b. "OWL-S: Semantic Markup for Web Services". <http://www.w3.org/Submission/OWL-S>.
- Wu, H., Li, Z., et al., 2010, Monitoring and Evaluating Web Map Service Resources for Optimizing Map Composition over the Internet to Support Decision Making, *Computers and Geosciences*, vol.37, no.4, 2011, pp485-494.
- Yang, C., Wu, H., et al., 2011. Using spatial principles to optimize distributed computing for enabling the physical science discoveries, *Proceedings of National Academy of Science (PNAS)*, 108(14), pp.5488-5491.
- Yue, P. et al., 2007. Semantics-based automatic composition of geospatial Web service chains. *Computer & Geosciences*, 33(5), pp. 649-665.

6. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant 41023001, 41021061 and 40971211.