Workshop of the ICA Commission on Map Generalisation and Multiple Representation – June 25th 2006

Establishing an OGC Web Processing Service for generalization processes

Theodor Foerster & Jantien Stoter

ITC, International Institute for geo-information and earth observation
P. O. Box 6, 7500 AA Enschede, the Netherlands
{foerster, stoter}@itc.nl

KEYWORDS: Generalization Web Service, automated web-based generalization processing, OGC, Open Source

1. Introduction

Exchanging knowledge in terms of sharing and reusing generalization algorithms was identified as a key requirement by the generalization research community to ease further research (Edwardes et al., 2003). Therefore a common mechanism as Web Services, which provides an open platform for world-wide access to different functionalities, was suggested to be promising. Web Services are distributed over the Web and provide certain functionality, i.e. generalization operators of various developers. Additionally the application of Web Services for web-based generalization processes would benefit from the established web-based data dissemination approaches, which are mostly implemented by Spatial Data Infrastructures (SDI). Hence Web Services enable on-demand and on-the-fly generalization processes based on the most current data. By the way, different commercial software solutions (e.g. ArcGIS) incorporate only basic generalization functionality but utilize Web Services, thus enabling web-based generalization processes would also increase their functionality.

However to apply Web Services as a means of knowledge exchange, these Web Services have to be accessible in an interoperable way based on common standards. Such a common standard (e.g. SOAP) has been applied by different research projects about web-based generalization processes (Neun and Burghardt, 2005; Neun et al., 2006; Regnauld, 2006). These projects gave already good insights into the capabilities of Web Services, but did not reflect the geospatial issues due to the missing geospatial concepts within SOAP (e.g. feature encoding). Thus some further research about evolving standards within the GI-domain has to be carried out.

Currently a geospatial processing service interface is getting standardized by the Open Geospatial Consortium (OGC), labeled as OGC Web Processing Service (WPS) (OGC, 2005). Since the generalization process is spatially related, it is relevant to investigate possibilities and limitations of the WPS specification with respect to requirements of generalization applications and research.

The following sections (2 & 3) will give a short insight into the WPS specification and its relevant features in terms of generalization processing. Section 4 presents the architecture of the developed WPS and shows how its implementation enables a research platform for generalization by addressing technical (extensible software architecture) and organizational (Open Source licensing) aspects. Sections 5 & 6 give an example of accessing web-based generalization functionality provided through the WPS and show the drawbacks of the current version of the specification. The last two sections describe the broader context of the related research and come up with a conclusion. All the presented results are part of an on-going research project at ITC's Department for Geo-Information Processing (GIP) (Poppe et al., 2006).

2. The WPS specification

The specification of the WPS (version 0.4.0) was released as a discussion paper (OGC, 2005). The notion of the specification is to provide spatial processes through a standardized service interface over the Web based on a common transfer protocol, namely the Hypertext Transport Protocol (HTTP). The variety of spatial processes that can be described by the WPS is unlimited. A process description for each process is available through the WPS interface. Besides a title and an abstract the description includes valid process parameters and their encoding. The client-service communication is based on the Extensible Markup Language (XML).

In detail the specified service operations of a WPS are *GetCapabilities* to provide service metadata, *DescribeProcess* to provide a complete description of the designated process and *Execute* to trigger the designated process (Figure 2).

3. Important features towards sufficient generalization processes

As generalization processes are mostly complex and consist of different sub-processes and generalization operators, integrating different Web Services to so called chains is an important feature. Thus the specification provides basic functionality based on HTTP-GET to access Web Services serving geospatial data as the Web Feature Service (vector data) within the WPS. A sophisticated approach towards triggering other Web Services via HTTP-POST and thereby to build even more complex chains is not yet supported.

Additionally the specification provides capabilities to execute long term processes without overstraining the capabilities of HTTP. This is an important feature, as generalization processes can be complex and thereby end up in time-consuming execution. Such a WPS is able to store the processed results and return them to the client on-demand. This avoids recalculation of time-consuming processes as well. However the client will not get notified instantly after the process is finished. In a worst case scenario this could result in a bottleneck of communication, because the client has always to check the status by itself. A more sufficient mechanism of notification would tackle this problem.

4. Architecture

To get a better insight into the notion of the WPS and to test its suitability as a research-platform for generalization, we decided to implement the specification as a proof-of-concept. Hence the suggested implementation had to be suitable for different generalization algorithms and data encodings. To meet these requirements we designed an extensible architecture (Figure 1). Such an extensible approach is realized by the notion of repositories, which provide dynamic access to the embedded functionality of the WPS. As the architecture has to be extensible for data handling and processing, repositories for these both issues were designed. Thus the three operations (*GetCapabilities*, *DescribeProcess* & *Execute*) interact with the incorporated functionality only via these repositories.

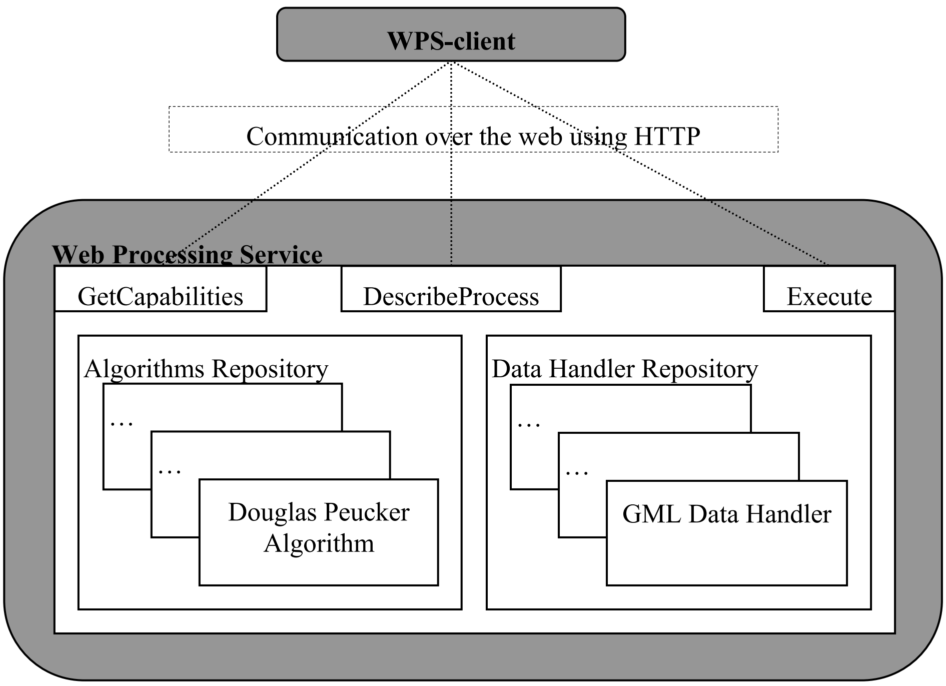

Figure 1. Architecture of the WPS

4.1 Implementation

The implementation is meant as a proof-of-concept to establish the WPS as a sufficient research platform for generalization. To demonstrate the nature of the WPS in a user-friendly way, we implemented also a WPS-client, based on an easy-to-use GIS as the Open Source GIS JUMP (www.jump-project.org). Using the WPS through JUMP provides the user to integrate the processed results into the user's further GIS tasks (e.g. mapping, spatial analysis).

To realize the extensible architecture of the WPS, the implementation features a plug-in mechanism, which provides functionality to query and execute the embedded functionality. Each data handler or process, which has to be accessible to the service, has thereby to implement a designated interface and has to be registered at the designated repository.

4.2 Open Source Licensing

The goal of the research was to establish a research platform, which is public accessible not only in terms of standardized Web Service interfaces but also in terms of open software use. Therefore we embedded the implemented prototype within the arising 52º North Open Source initiative (Kraak et al., 2005; www.52north.org). The implementation is thereby available under the GPL license.

5. WPS in action

To demonstrate, how to carry out generalization processes using a WPS implementation, we assume that we want to apply a simplification algorithm as the Douglas-Peucker algorithm (Douglas and Peuker, 1973) to some road data. Therefore we have to access the WPS and walk through the common client-service communication, as illustrated in the sequence diagram (Figure 2).

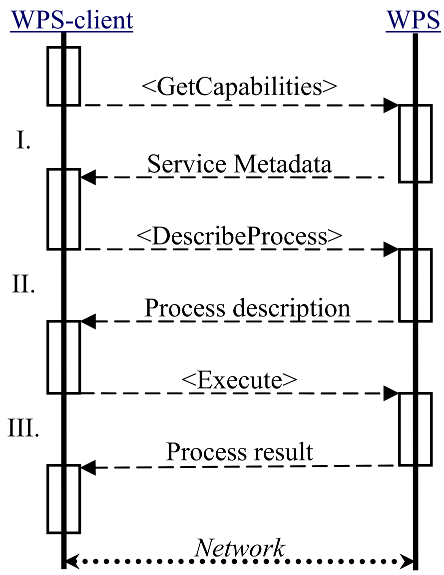

Figure 2. Basic client-service communication

In order to use the WPS we have to pass the URL of the WPS to the client at first. The client will retrieve the service metadata through the *GetCapabilities* operation (step I.). Thus we discover that the WPS provides a process which is called Douglas-Peucker algorithm. Hereafter the client retrieves the process description for the Douglas-Peucker algorithm via the *DescribeProcess* operation (step II.). Based on the retrieved process description the client is able to generate a form as demonstrated in Figure 3. The Douglas-Peucker algorithm will be triggered with the required parameters (tolerance value: 150 meters) and the original data by the *Execute* operation (step III.). Thereafter the WPS returns the processed result.

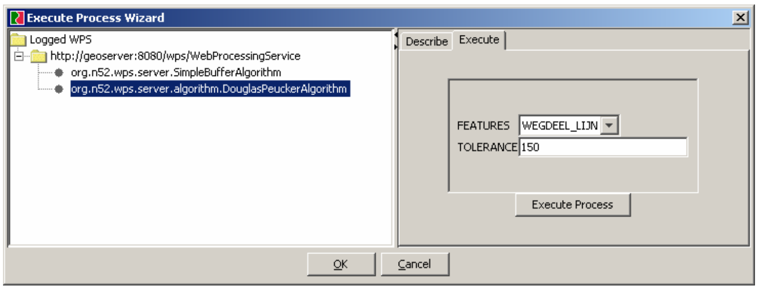

Figure 3. Sample form to configure a generalization process

The original data and the result performed by the Douglas-Peucker algorithm, which is provided by the WPS, are shown in Figure 4 & Figure 5. Although there are some errors produced by the algorithm due to missing topology checks, the short example shows that the WPS is suitable to perform generalization processes straight-forward.

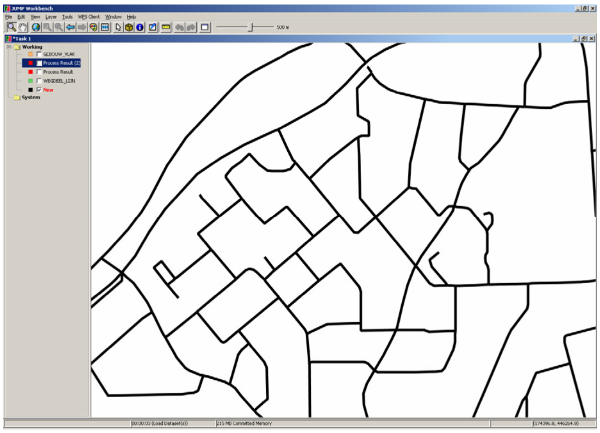

Figure 4. Original road data

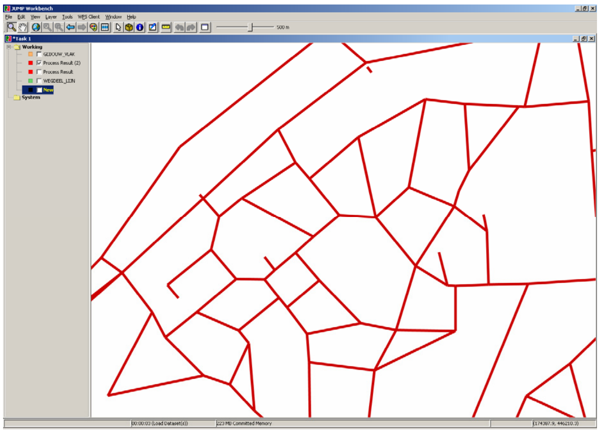

Figure 5. Simplified road data

6. General remarks

As mentioned before the WPS does not provide sufficient communication mechanisms. Let's assume, that we want to process not only a few street segments as in the example, but a complete tile of road data. That would result in a huge amount of data to be transferred over the Web and to be processed at the WPS and thereby would lead to a significant delay. Regarding the current version of the WPS, the client had to stay idle or had to re-check the status over and over. Both approaches end up in a waste of computational resources. A mechanism enabling the WPS to notify the client instantly right after the process is finished, would be more suitable.

A possible solution to provide a more sufficient communication mechanism, would be to utilize a notification service, as specified in OGC (2003), which would notify the client via e-mail, SMS or just by a simple HTTP-callback. Incorporating such an existing specification as the Web Notification Service (WNS) would result in small changes in the WPS specification itself, but would lead to an improved and open architecture for sufficient and scalable processing.

In general the possibility to integrate different processes supplied by a WPS or other complex Web Services (e.g. Web Coordinate Transformation Service) within a chain is not yet possible. This is due to the fact, that the specification does not provide any mechanism to include other Web Services within the *Execute* operation, which are accessible through HTTP-POST. Therefore the encoding for the Execute operation has to be adjusted slightly. This would improve the specification significantly and would enable more complex processing within a WPS.

The most essential problem of the WPS from our perspective is the lack of concepts to provide semantics, to establish transparent processes. Looking on the example, it becomes clear, that it is not obvious, what type of process is carried out by the name “Douglas-Peucker algorithm”. For cartographers the meaning of “Douglas-Peucker algorithm” and its parameters might be clear, but regarding an approach, in which an algorithm has to be applied within a computational environment automatically, the meaning would be unperceivable by the name of the process. The name of a process is not unique and does not provide sufficient transparency to the client.

Thus a mechanism is required to enable computational environments to apply such processes meaningfully. A possible approach is to come up with so called service profiles. Such a service profile specifies the domain-specific processes and provides a mechanism to maintain the uniqueness of processes. This approach is just evolving within OGC and led to some discussion, which resulted in a specific WPS service profile for Coverage Web Processing Services (OGC, 2006). Also the generalization community proposed such service profiles, by introducing a classification for Web Service for generalization (Burghardt et al., 2005).

As the meaningful description of Web Services is not only a problem within the OGC but also in the web community in general, concepts related to ontologies were carried out, which led to the definition of semantic description of data (RDF) and semantic description of web services (OWL-S). These concepts will contribute fruitfully to OGC Web Services, but require further investigation (see also (Lemmens, 2006)).

7. Outlook

To give a deeper insight into the related PhD research project at GIP, we would like to introduce our future plans shortly. The research focuses on the development of a concept towards automated web-based generalization processes. To carry out web-based generalization processes the generalization operators, which build the basis for each generalization process, have to be accessible meaningfully. Therefore formalization as a means to establish meaning in a computational environment has to be applied to the generalization operators.

Thus the main objectives of the research are to formalize the generalization operators, to embed this formalization in a web-suitable approach (OWL-S) and to build a meaningful architecture for automated web-based generalization processing. This architecture will be applied to an existing generalization software solution as a proof-of-concept. In our case the WPS provides basic facilities for standardized processing (i.e. generalization operators), which will be utilized within the designated architecture.

8. Conclusion

The paper showed that our approach towards a web processing service for generalization - based on OGC standards - led to promising results (as also demonstrated by other projects (Ostlaender et al., 2005)). The resulting prototype can be customized with different generalization algorithms and data-specific handlers. Additionally the prototype is available under Open Source license. Both, our technical approach and our organizational approach will contribute to improve the knowledge exchange within the generalization community.

The prototype gave also insight into the WPS specification. Thus it became clear, that there are three requirements for a generalization web service, which are not yet covered by the current specification:

- Semantically enriched description of processes
- Sufficient communication patterns
- Chaining of processes

Some solutions have been presented within the paper.

Additionally the paper introduced the objectives of the on-going PhD research project at GIP related to automated web-based generalization processes. Since the meaning of generalization operators is extremely important in that context, applying the concepts of the semantic web is promising. Overall establishing formalized generalization operators within a web-based environment will lead to a meaningful architecture for automated web-based generalization processing.

Therefore the WPS seems to be suitable to provide a valid basis, because it enables standardized access to generalization operators and ensures thereby the syntactic interoperability on a basic level for web-based generalization processing sufficiently.

References

Burghardt, D., Neun, M. and Weibel, R. (2005). Generalization Services on the Web - A Classification and an Initial Prototype Implementation. In *Proceedings of Auto-Carto* 2005, Las Vegas, Nevada.

Douglas, D.H. and Peucker, T.K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer, 10*, pp.112-122.

Edwardes, A., Burghardt, D., Bobzien, M., Harrie, L., Reichenbacher, T., Sester, M. and Weibel, R. (2003). Map Generalization Technology: Addressing the Need for a Common Research Platform. In *Proceedings of ICC 2003,* Durban, South Africa.

Kraak, M. J., Sliwinski A., and Wytzisk, A. (2005). What happens at 52N? An open source approach to education and research. In *Proceedings of Joint ICA commission seminar 6-8 July 2005, part of the 22nd ICA conference, ICC 2005: Mapping approaches into a changing world*, A Coruna, Spain.

Lemmens, R. L. G. (2006). *Semantic interoperability of distributed geo-services*. PhD Thesis, Delft, the Netherlands, Delft University of Technology.

Neun, M. and Burghardt, D. (2005). Web Services for an Open Generalisation Research Platform. In *Proceedings of 8th ICA Workshop on Generalisation and Multiple Representation*, A Coruna, Spain.

Neun, M., Burghardt, D. and Weibel, R. (2006). Spatial Structures as Generalization Support Services. In *Proceedings of Workshop on Multiple Representation and Interoperability of Spatial Data*, pp. 6-15. Hannover, Germany.

OGC (2003). *OGC Web Notification Service*. OpenGIS Discussion Paper. Simons, I. and Wytzisk, A. (eds.).

OGC (2005). *OpenGIS Web Processing Service*. OpenGIS Discussion Paper. Schut, P. and Whiteside, A. (eds.).

OGC (2006). *OpenGIS Web Coverage Processing Service*. OpenGIS Discussion Paper. Baumann, P. (ed.).

Ostlaender, N., Tegtmeyer, S. and Foerster, T. (2005). Developing an SDI for time-variant and multi-lingual information dissemination and data distribution. In *Proceedings of EC-GI & GIS Workshop, ESDI: Setting the Framework*, Alghero, Sardinia.

Poppe, E., v. Elzakker, C. P. J. M. and Stoter, J. E. (2006). Towards a Method For Automated task-driven Generalisation of Base Maps. In *Proceedings of UDMS 2006 - 25th Urban Data Management Symposium*, pp. 3.51-3.64, Denmark, Aalborg.

Regnauld, N. (2006). Improving Efficiency for Developing Automatic Generalization Solutions. In *Proceedings of ISPRS Workshop: Multiple Representation and Interoperability of Spatial Data,* pp. 1-5, Hannover, Germany.