Towards automatic web-based generalisation processing: a case study

Theodor Foerster, Jantien Stoter & Rob Lemmens
{foerster,stoter,lemmens}@itc.nl
International Institute for Geo-Information Science and Earth Observation, P.O. Box 6, 7500 AA Enschede, the Netherlands

Web Generalisation Services (WGS) play a big role in current generalisation research. However they are still lacking interoperability which is required for interchanging them. Their service interfaces are currently not generic and mostly reflect just the specific interface imposed by characteristics of the algorithm they provide. The missing interoperability limits automatic generalisation processing on the Web. In this paper we first describe our view on what WGS comprise. In addition we propose a methodology to increase the interoperability of WGS applying common concepts of the Web as XML-schema and namespaces. We demonstrate the effect of generic interfaces for generalisation operators by introducing a case study that is based on ratio-based simplification.

Keywords: Web Generalisation Service, interoperability, profiles, ratio-based simplification.

1 Introduction

Web Services are used in various domains such as transportation or tourism to provide different functionality over the Web. In the context of generalisation research the group of the University of Zürich has coined the term of *generalisation service*[1] (Burghardt et al., 2005; Neun & Burghardt, 2005). This special breed of Web Service provides functionality to fulfill certain tasks of generalisation processing. The group also introduced a hierarchy of generalisation services based on a high-level and coarse-grained classification of different interaction patterns for generalisation processes. This classification of generalization services enhances the automated and meaningful use of generalization services, as the classification helps to identify the intended use of generalization service instances. Thus the hierarchy classifies process, operator and support services (see also Section 2.2).

When looking at Web Services in general, the interoperability of Web Service towards meaningful Web Service interaction is still a problem. This is also the reason for missing capabilities of (semi-) automatic processing on the Web using generalisation services. Semi-automatic processing is a key requirement for web-based generalisation, but also to improve the acceptance of a web-based approach within the generalisation community.

Nowadays the interoperability is ensured by interface descriptions on the service level (e.g. Web Service Description Language (WSDL)), which act as a contract for basic Web Service communication, but prevent a meaningful interaction with the actual functionality of the Web Service. Therefore a contract on the process-level has to be established. This so called *process-level interoperability* also provides better transparency for evaluating different generalisation services and their produced results. Finally it improves application development, as the interfaces are well-known and implementation-independent not only at the service level but also on the process level.

A contract on the process level can be established by specific Web Service interface descriptions – so called profiles. These profiles are defined and used within a community by reflecting the common understanding (see Section 3). We are aware of the fact that the terms interface and profile or specific and generic can be used in the same context. However, we want to clarify for this paper, that an interface provides an abstraction of the generalisation functionality and is thereby generic. Otherwise a profile is a restriction (from service to process level) of a Web Service description and thus specific[2].

In the context of generalisation such profiles could be appropriate for Web Services, which provide generalisation functionality by means of generalisation operators. According to the generalisation service hierarchy such services are labeled as *operator services*. This paper focuses therefore on operator services and on improving their interoperability. In the future process and support services should be considered as well to improve automation of the whole generalisation process.

We propose an approach for defining and deploying profiles for operator services by using the concept of XML-schemas and namespaces. These concepts address two issues: Firstly the XML-schemas describe the structure and the data types of the service interface and secondly the namespaces

[1] This term can be used interchangeably with Web Generalisation Service (WGS).
[2] This also exposes a trade-off for generalisation research on the Web that some abstractions of operators might provide a less specific basis for Web Services.

solve the conflict of heterogeneous naming of operators (i.e. naming of the same thing differently or using the same name for different things). As we design profiles for operator services, it is possible to interoperate meaningfully with the operator services belonging to the same profile, but also the generalisation algorithms (of the same operator) become interoperable according to the generic interface. Actually the second aspect improves the transparency of generalisation algorithms regarding their process results, because different algorithms can be executed meaningfully based on the same parameters and parameter values. This improves the comparability of such algorithms and their process results.

To demonstrate the applicability of our approach we introduce a case study, namely a profile for simplification based on a ratio-based approach (Foerster et al., 2007). It is important to note, that our approach of profiles subdivides the category of operator services (Burghardt et al., 2005).

The paper first introduces Web Services and the special case of generalisation services by also revealing our view based on common definitions and recent literature on web-based generalisation (Section 2). Section 3 demonstrates an interface on the example of simplification and describes the role of profiles within the context of interoperability. The last section (Section 4) raises some open issues and ends with a conclusion.

2 Web Services for Distributed Generalisation Processing

The following part of this section provides basic definitions about Web Services and interoperability, but also introduces the idea of Geospatial Web Services. The next part of this section presents an overview of recent research and implementation activities on generalisation services. After that the section suggests a modified hierarchy of generalisation services, which seems to be more suitable for complex scenarios of web-based generalisation processing. As the paper focuses on operator services we want to suggest finally in this section their main design characteristics.

A Web Service is a software component, which provides functionality through a web-accessible interface in a platform- and programming-language independent way. Furthermore the Web Service interface is described in a computer-understandable way and mostly encoded in WSDL (Gottschalk et al., 2002). The understanding of the Web Service interface description is a fundamental requirement to ensure interoperability.

Interoperability is the capability of two components (services) to communicate at run-time in order to meet a common goal. ISO 19119 identifies two levels of interoperability for Web Services (ISO/TC211, 2005):

- Syntactical – the Web Services use the same structure and input/output format for the information.
- Semantic – the Web Services communicate based on an agreed meaning of their parameters.

The introduced terms of service-level and process-level interoperability address the syntactic interoperability, each one at a different level of granularity. However after the profiles are integrated manually into the Web environment, profiles enable a meaningful Web Service interaction. The process-level interoperability has then the same characteristics as the semantic interoperability, but achieved these characteristics only in a semi-automatic way.

Common Web Service technology ensures the interoperability through mechanisms such as XML-schema and namespaces. XML-schema describes the structure of any XML-document. Namespaces in this context provide a mechanism to identify the equality of the XML-schemas, which are linked by different XML-documents.

A special breed of Web Service is the Geospatial Web Service. Its functionality is dedicated to the geospatial domain (e.g. mapping, geodata provision and geo-processing). The Open Geospatial Consortium[4] (OGC) and the ISO/TC 211 Geographic information/Geomatics[5] specify the functionality of such Geospatial Web Services. It is important to note that the specifications of specific Geospatial Web Services (map service, feature service) provide syntactical interoperability and also a certain degree of semantic interoperability, as the meaning of the messages and their parameters are documented by the specifications. However, this meaning is not sufficient to enable full-automatic Web Service interaction. But as these specifications are designed for a special purpose (e.g. map

[4] http://www.opengeospatial.org
[5] http://www.isotc211.org/

portrayal or feature retrieval), they provide more precise semantics (i.e. process-level interoperability) than common W3C Web Services.

We also would like to mention current efforts of OGC towards a Geospatial Web Service for distributed geospatial processing, labeled as Web Processing Service (WPS) (OGC, 2005a). The WPS provides a simple way to web-enable processes using Web Service technology. It is simple, as it provides a straight forward communication pattern, which involves three operations, which are requested over HTTP: *GetCapabilities*, *DescribeProcess* and *Execute*. All these operations return XML documents, containing service or process metadata or process results. It is thereby more abstract than WSDL and less complex, as the requests and responses are not that complex as they can be in WSDL. The WPS approach has been already successfully applied in Foerster & Stoter (2006) for generalisation services, but it only provides interoperability on the service level.

However we want to stress, that WPS is not the focus of this paper nor can it be used for chaining different services towards a super-WPS. Chaining of services should accomplished by using BPEL (Business Process Execution Language). Therefore we propose that WPS is applicable to atomic operator services as described in Section 2.3.

2.1 Generalisation Services

Burghardt et al. (2005) introduced a three level hierarchy of generalisation services, which should improve the semi-automatic and meaningful interaction with generalisation service instances. They attached *service interfaces* and *interaction interfaces* to indicate the possible ways of interaction. At the bottom level of the hierarchy there are simple *support services*, which provide basic GIS-functionality through a service interface for enriching the data with information needed for the generalisation process (such as triangulation). The next level is the *operator service*, which provides generalisation functionality through a service interface and an interaction interface. An operator service can utilize functionality of support services. The operator service follows the basic concept of the generalisation operator, which is derived from the first experiments with automated generalisation (McMaster 1992). Generalisation operators encapsulate atomic – also called stand-alone - generalisation functionality and provide an abstract view on the generalisation algorithms that implement such operator. Atomic in this context means that the operator does not interfere with other operators. Although there are several proposals to classify generalisation operators (see for example McMaster (1992), and Bader et al. (1999)) there is not yet a classification that defines all available operators unambiguously. Some related issues to such a classification will be discussed in Section 3. Both the operator service and the support service are used by the *process service*, which can actually drive the generalisation process and automatically evaluate the results. In Neun and Burghardt (2005) the process service only has an interaction interface attached, thus it is not possible to interact with this type of service in an automatic way.

Burghardt et al. (2005) gave also an overview of the evolution of generalisation services until the year 2005. Since then specification programs and research drew more attention on generalisation services especially on operator services. When OGC proposed the WPS specification, Foerster & Stoter (2006) implemented it as a platform for generalisation services and made it available under Open Source license at 52º North[6] as a web-based processing service framework (Foerster 2006). Regarding the generalisation service hierarchy of Burghardt, the framework is applicable to all types of generalisation services.

Harrower and Bloch (2006) developed an interactive web application to demonstrate the effects of generalisation (i.e. simplification). They also labeled their application as Generalisation Service. Concerning the generalisation service hierarchy they developed an operator service with an interaction interface. Related to Geospatial Web Service development van Oosterom et al. (2006) proposed an extension of the Web Feature Service interface for progressive transfer. An extension of the Web Feature Service for multi-scale data dissemination was also proposed by Hampe & Intas (2006).

Until now the services are used for single remotely performed operations, but not chained to perform complex generalisation functionality involving multiple services and operators. This is mostly due to the fact that the interfaces of the services do not provide full interoperability or that the tools to chain different generalisation services are not available. Additionally, current generalisation services can only be accessed by tools using a graphical user interface to configure the process manually by inspecting the parameters of the algorithm (Figure 1). This means that these tools always require human reasoning based on sometimes poor descriptions of remote generalisation algorithms.

---

[6] http://www.52north.org

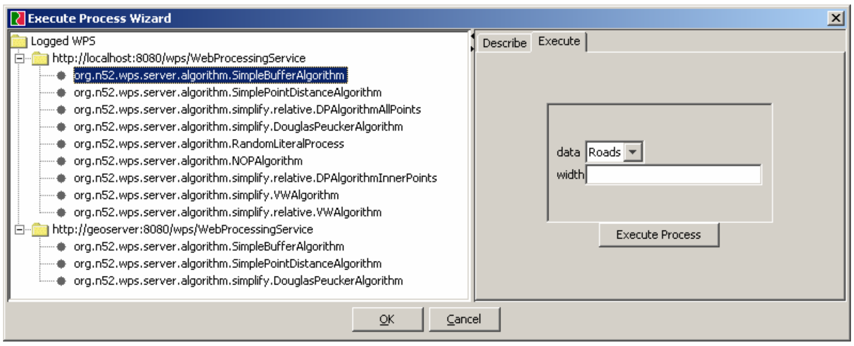

Figure 1: Form for manual interaction with remote generalisation algorithms (WPS Client, www.52north.org).

2.2 Discussion of the Current Generalisation Service Hierarchy

Generalisation Services in our perspective should a) access generalisation functionality in a distributed way by chaining multiple services and should b) provide ideally automatic generalisation functionality to other Web Services. The hierarchy of generalisation services does not cover these requirements sufficiently. Operator services should be able to perform generalisation functionality by accessing also process services (requirement a). Such process services can provide capabilities to control and evaluate complex service chains, which could support for instance an operator service providing the amalgamation of complex polygons based on an optimization approach for cartographic representation. The internally applied service chains would not be visible through the interface of the operator service to the outside world. This is also known as opaque chaining.

We propose to extend the generalisation service hierarchy and to add also a service interface to the process service and to rearrange the dependencies in a more open classification (Figure 2 changes to the original figure are indicated in bold). Attaching a service interface to the process service would also expose it as a Web Service and add more consistency in wording according to the definition of Web Services. Furthermore a service interface for process services would enable Web Services from other domains to access generalisation workflows automatically (requirement b).

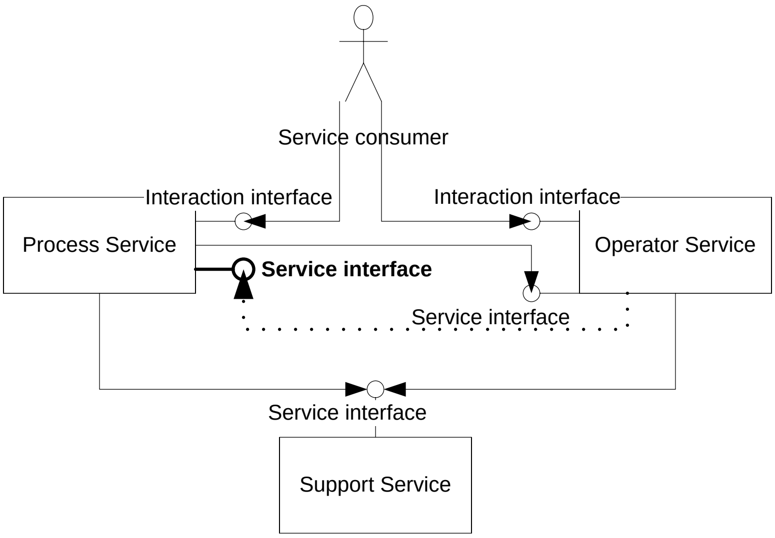

Figure 2: Modified classification of generalisation services – modification: attached a service interface to the process service (indicated as **bold**) and the operator service can depend on a process service (dotted line)) (modifications based on (Neun & Burghardt 2005)).

The subdivision of operator services according to the addressed data type as proposed by Burghardt et al. (2005) is not needed, since the interface of each operator service itself can advertise a specific data schema (i.e. complex data type) of the provided generalization functionality. The data type of the input and output parameters of such generalization functionality could be additionally described by means of ontologies in order to ensure the semantics of the parameters.

2.3 Design Suggestions for the Operator Service

Besides the important role of data types for an appropriate description of any generalisation service interface, an operator service should meet more specific requirements.

The operator service should be a light-weight component, which provides stateless access to generalisation algorithms that implement the specific operator. Consequently the interaction with the operator service should be based on a single request. The reason for this design suggestion is that the operator inherits an atomic function, which is quite easy to handle for the client[7]. Thus the operator service (as well as the client) does not have to handle complex request procedures with multiple requests.

Additionally the request should contain the process data directly or a link to it. The response should also contain the processed result (again either directly or as link). This avoids complex data handling mechanisms incorporated inside the operator service and ensures the light-weight character of operator services. Mechanisms for transaction and session handling are also not required using this approach.

Summarising, an operator service should be a Web Service, which:

I. Provides (atomic) functionality of a generalisation operator by also using process and support services in an opaque fashion (Section 2.2).
II. Is stateless
III. Consumes the necessary process data through the interface
IV. Includes the processed result in the response. However an operator service does not provide any mechanism for session handling (this complies with point II.).
V. Advertises the applicable data type in its interface description, as precise as possible.

Although these properties are abstract, they guide the design of the operator service and help to assess the design quality of existing and future operator services. We propose to apply for operator services the WPS specification, as it allows implementing atomic functionality in an easy way.

We are aware of the fact, that atomic Web Service functionality can result in a huge communication overhead, however we propose, that the client has to apply smart communication and caching capabilities to limit the communication overhead with such operator and support services to an applicable level. Especially caching is promising, as any resource on the web can be identified by its URL. So in a production environment, which applies operator services, most of the data and generalisation results have only to be performed once, but can be served multiple times even over different use cases. Thus the communication will be reduced to the required maximum.

3 Profiles for Operator Services

After presenting our view on generalisation services, this section describes our approach towards automatic web-based generalisation processing. Thus this part of the section presents the general process to define the generic characteristics for the generalisation operator. Then the characteristics are applied to the design of the interface and finally the last part of the section proposes an architecture, which incorporates the profile and demonstrates how it enhances the automation of web-based generalisation processing and improves the process level interoperability.

Our proposed approach for designing profiles for operator services is demand-driven. First the demand of the scenario, which prescribes when and what objects to generalise, guides the selection of the appropriate operator. Then based on the demand the generic interface for the chosen operator has to be designed. We are aware of the fact that different demands may lead to the same operator but result in different profiles. So there might evolve multiple generic interfaces for the same operator. However, to optimize interoperability only some major interfaces for one operator should be developed (as it will be demonstrated by the case study).

A first source to select the appropriate operator is the classification of McMaster. However this classification does not reflect the more product-oriented approach of web-based generalisation processing. Web-based generalisation processing has to support the dissemination of derived data according to a desired data model (level of detail) or of derived map (at another scale). This two-fold requirement fits with the Grünreich model and its classification into model and cartographic generalisation (Grünreich, 1992). In Foerster et al. (2007) we propose therefore a new classification of operators, according to this model and describe the main generalisation operators based upon standardized models for data (ISO/TC211, 2003) and cartographic representation (OGC, 2005). As this operator classification is based on standardized models, which could be converted to ontologies[8], this

[7] Please note, that a client can be either a human user or a Web Service.
[8] As presented here: http://loki.cae.drexel.edu/~wbs/ontology/iso-19109.htm (accessed June 2007).

mentioned paper takes a step towards a formalization of these operators and thereby adds a new aspect to generalisation research.

3.1 The Design of the Ratio-based Simplification Interface

This case study shows the potentials of operator services by implementing simplification as an operator service. Thereby simplification in our case aims at the major demand of data reduction. We assume that simplification is a part of model generalisation as it is mainly used to derive a data set with less detail. This assumption is adopted from literature such as Saalfeld (1999).

Following the demand of data reduction we propose a *ratio-based simplification*, as described in Foerster et al. (2007). The simplification ratio basically compares the number of nodes before and after the simplification process. As mentioned in Foerster et al. (2007) this measurement is not sufficient for cartographic simplification, but cartographic aspects were considered as minor concern in the introduced case study.

The interface for a ratio-based simplification should reflect two main issues. It should advertise the simplification ratio and additionally it should be limited to a feature model-based representation, in order to explicitly exclude the cartographic aspects. Those can only be addressed sufficiently if the data model is combined with a cartographic model.

A common feature model for geospatial data is described in the General Feature Model (GFM) of ISO 19109 (ISO/TC211, 2003) and broadly used and implemented in the context of Geospatial Web Services. The interface description for the simplification operator includes this model to reflect the vector-based approach of the ratio-based simplification. The simplification ratio is of type double and ranges from 0.0 for no simplification to 1.0 for full simplification (see Figure 3). A full simplified data set of roads only consists of edges linking the start- and end-nodes of the original data set (i.e. intermediate nodes were removed).

It is important to note, that the ratio-based simplification approach could be applied to any simplification algorithm, which provides a ranking of points of a line such as the Douglas-Peucker (Douglas & Peucker, 1972) and Visvalignam-Whyatt (Visvalingam & Whyatt, 1993) algorithm do, which are applied in the presented case study (Section 3.2). Based on such a ranking it is possible to apply a ratio and to sort out the unimportant points.

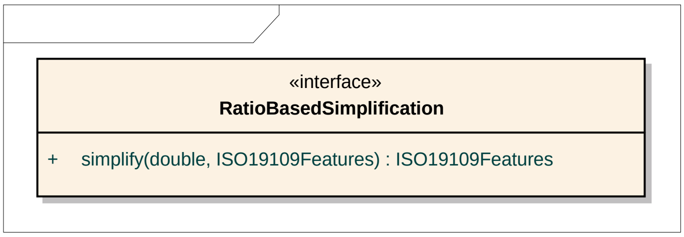

Figure 3: Proposed interface for the ratio-based simplification in UML notation.

3.2 Deploying the Profile for Ratio-based Simplification

To sufficiently deploy generic interfaces as the proposed interface for ratio-based simplification and to benefit from the concept of namespaces, we propose to describe the profile of the operator service with XML-schema. This enables every service provider or client to directly validate the message exchange on the process level.

The profile for the ratio-based simplification (Figure 4) links the XML-schema of the Geographic Markup Language (GML version 3.1, specified in ISO 19136) to represent the GFM and thereby the vector-based process data. The simplification ratio incorporated in the profile is based on XML-Schema's basic type double (as described in Section 3.1).

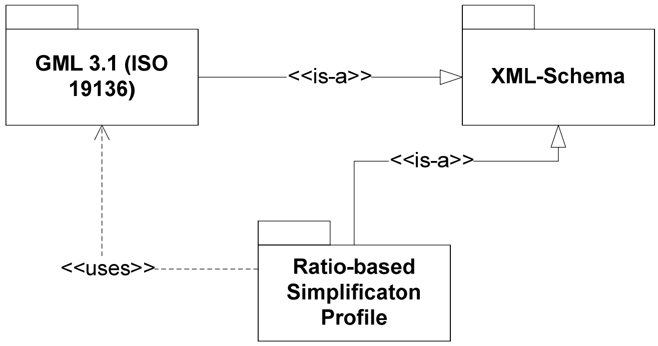

Figure 4: Dependencies of the ratio-based simplification profile.

To install the designed XML-schema, it is located as a web-accessible resource and linked to a namespace. This namespace can be used by any Web Service or client, which respectively provides or wants to access such ratio-based simplification. The concept of namespaces is the key technology to handle naming in distributed systems such as the web.

The concepts of Web Service technology allow clients and Web Services to automatically identify (namespaces) common knowledge and to communicate (XML-schemas). But in order to do that the knowledge (i.e. the parameters and the semantics) about the ratio-based simplification operator has to be incorporated manually once. This process can range from simple interface design towards complex integration into already existing application environments. Thus the proposed concept of profiles remains to be still semi-automatic. However after this pre-configuration phase is passed, every generalisation algorithm, which implements the profile can be linked automatically into the pre-configured system.

The example in Figure 5 demonstrates the described approach based on a Web Service architecture. It examines the manual and the automatic part of the approach but also demonstrates the improved interoperability. So at first the service providers have to take care that their generalisation algorithms (in our case Douglas-Peucker algorithm and Visvalingam-Whyatt algorithm) match the profile syntactically and semantically. Otherwise the algorithms cannot produce satisfying results. In this phase, the providers already apply the abstraction on the algorithms towards the generic interface (Section 3.1). Finally their services have to refer to the namespaces and implement the service interface according to the profile.

The client also has to know the syntax and the semantics of the profile. This preparation is the manual part and involves human interaction for the client and the providers. At this stage, the client and the providers are referring to the profile. They both interoperate on a process level (looking from a Web Service view) as well as on an operator level (looking from the generalisation perspective).

As this stage is reached and the knowledge is incorporated about the ratio-based simplification at the client and provider side and refers to the namespace, the client is able to identify and to communicate automatically with the operator services. The client can assess the functionality by accessing the service descriptions of both operator services. As the services refer to the same namespace in their description it is possible for the client to identify that they (the client as well as the two operator services) rely on the same ratio-based simplification profile. Now the client is able to perform the desired generalisation operator on both services automatically by applying the same parameters as well as the same parameter values (i.e. the process results are comparable). Would the service description only identify the operator with non-standardized wording (e.g. *VWsimplify* and *DPsimplify* as in Figure 5), and some arbitrary non-standardized schema descriptions, no automatic interaction would be possible and human interaction/reasoning would be required (as it is nowadays the case in all environments for operator services).

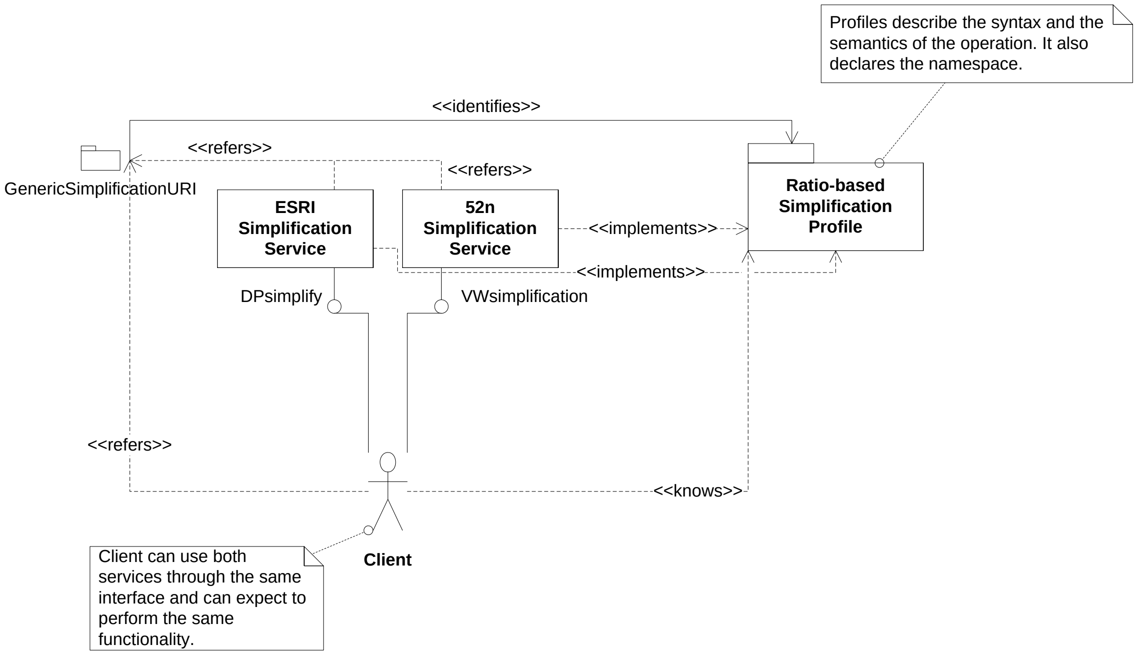

Figure 5 Semi-automatic generalisation processing based on namespaces and XML-schemas.

4 Discussion & Conclusion

This paper demonstrates by a case study on simplification, how operator services might become interoperable and finally how the automated web-based generalisation processing might be enhanced. This is achieved conceptually in two ways: By designing generic interfaces of generalisation operators based on a common demand (i.e. from algorithm to operator level, Section 3.1) and by establishing profiles (i.e. restriction from the service level to the process level, Section 3.2). Finally it is possible to perform automatic and on-the-fly web-based generalisation processing, although initial pre-configuration is required. The automation on the web is realized by XML-schema and its concept of namespaces. The combination of all these aspects improves the interoperability of operator services. Especially process services of generalisation can benefit from such an increased interoperability as they are now able to automatically access different operators and to evaluate on-the-fly the result provided by different operator services meaningfully.

Operator services in general have demonstrated to be the backbone for web-based generalisation processing. The interface design of such an operator should always be demand driven. In the case of simplification this approach led to the simplification ratio. The combination of this ratio and the restriction on the data type (i.e. using ISO 19109) allowed building an unambiguous interface.

The introduced case study of ratio-based simplification is simple, but shows sufficiently, that the concepts are appropriate. Even the simple interface incorporating only two parameters is sufficient and a more complex scenario involving a higher number of parameters would not lead to a more sufficient result. However the case study has to be applied to a more sophisticated Web Service scenario of publish-find-bind (Gottschalk et al. 2002) in order to fully proof the approach. Additionally such scenario should involve more different operator services, which are chained to a complex generalisation functionality. Also the extraction of more generic interfaces for other operators should be carried out in this context. Finally such profiles should be made available and maintained by a responsible party within the generalisation community (such as ICA). Then every provider and client could rely on this repository in order to incorporate and provide generalisation functionality in an interoperable way on the Web.

Regarding the profile definitions, the semantics of the thematic aspects of the input data and of the processed result are still not explicit. Thematic aspects become relevant for generalisation processing, if this information is needed in the generalisation process (either to indicate conflicts or as input for the operators). Formal ontologies describing all aspects of the data explicitly could therefore provide a more appropriate solution. Until then only operators are reliable that address the geometric aspects of the data. A more formal and semantically-aware approach towards web-based generalisation functionality could be adopted from Lemmens (2006), who proposed a concept of formalizing geospatial processes on the web. A concept for a comprehensive formalization of generalisation operators remains an aspect for research.

Regarding the achieved interoperability, applying profiles to Web Generalisation Services is not satisfying overall as manual interaction, at least in a pre-configuration stage, is required (Section 3.2). However profiles provide a way to at least identify the equality of different services and make thereby generalisation services comparable (based on namespaces). The current specification development of WPS also includes some mechanisms for enhanced process-level interoperability by incorporating the notion of process-profiles based on URNs (Uniform Resource Name). This is a more centralized approach, in which organizations have to propose a set of profiles, which are then identified via such a URN. However a URN is not able to be linked to an explicit XML-schema document and does not thereby provide further (syntactic) validation of the process. This approach has to show in future its applicability for interoperable processes on the Web.

In the case of an implementation of a more complex Web Service architecture the proposed requirements for operator services (Section 2.3) could be checked. A more complex Web Service architecture, which provides automatic web-based generalisation processing especially for other domains, could demonstrate, if the modified hierarchy of generalisation services (Section 2.2, Figure 2) is applicable.

5 References

Burghardt, D.; Neun, M. & Weibel, R.: Generalisation Services on the Web - A Classification and an Initial Prototype Implementation, *Auto-Carto 2005.*

Bader, M.; Barrault, M.; Regnauld, N.; Mustiere, S.; Duchene, C.; Ruas, A.; Fritsch, E.; Lecordix, F. & Barillot, X., AGENT Workpackage D2 - Selection of Basic Algorithms, *Department of Geography, University of Zurich, **1999.***

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

Foerster, T.: An open software framework for Web Service-based geo-processes, *Free and Open Source Software for Geoinformatics,* **2006.**

Foerster, T. & Stoter, J.: Establishing an OGC Web Processing Service for generalisation processes. *ICA workshop on Generalisation and Multiple Representation*, **2006**.

Foerster, T.; Stoter, J.; Koebben, B. & van Oosterom, P.: A Generic Approach to Simplification of Geodata for Mobile Applications, *AGILE 2007.*

Foerster, T.; Stoter, J. & Koebben, B.: Towards a formal classification of Generalization operators, *ICC, Moscow, Russia,* **2007**.

Gottschalk, K.; Graham, S.; Kreger, H. & Snell, J.: Introduction to Web service architecture, *IBM Journal,* **2002**, *41*, 170-177.

Grünreich, D.: ATKIS - a topographic information system as a basis for GIS and digital cartography in Germany, *From digital map series to geo-information systems,* **1992**, *Geologisches Jahrbuch Reihe A*, 207 216.

Hampe, M. & Intas, S.: Extension of the OGC Web Feature Service standard for multiple representation data, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* **2006**, *XXXVI / TC II*, 49-54.

Harrower, M. & Bloch, M.: MapShaper.org: A Map Generalization Web Service, *IEEE Computer Graphics and Applications,* **2006**, 22-27.

Lemmens, R. L. G.: Semantic interoperability of distributed geo-services, *Delft University of Technology,* **2006**.

ISO/TC 211: Geographic information - Rules for application schema, *ISO,* **2003**.

ISO/TC 211: Geographic information – Services, *International Organization for Standardization,* **2005**.

McMaster, R.B. & Shea, K.S.: Generalization in Digital Cartography, *American Association of Geographers,* **1992**.

Neun, M. & Burghardt, D.: Web Services for an Open Generalisation Research Platform, *8th ICA Workshop on Generalisation and Multiple Representation,* **2005**.

OGC, GO-1 Application Objects, *Open Geospatial Consortium, Inc.*, **2005**.

OGC: OpenGIS Web Processing Service, *Open Geospatial Consortium*, **2005a**.

van Oosterom, P.; de Vries, M. & Meijers, M.: Vario-scale data server in a web service context, *Workshop of the ICA Commission on map Generalisation and Multiple representation*, **2006**.

Visvalingam, M. & Whyatt, J. D., Line generalization by repeated elimination of points, *Cartographic Journal*, **1993**, *30*, 46-51.