Originally published as

Haubrock, S.; Theisselmann, F.; Dransch, D. (2007): Service-Oriented Architectures for Natural Disaster Management., CIG/ISPRS: Proceedings of the Joint CIG/ISPRS Conference on Geomatics Solutions for Disaster Management, CIG/ISPRS Joint Conference (Toronto, Ontario, Canada 2007), CD, 5.

Service-Oriented Architectures for Natural Disaster Management

Sören Haubrock, Falko Theisselmann, Doris Dransch

Abstract

Spatial information deduced from environmental monitoring devices, simulation models and scientific information sources are crucial for rapid response in the context of natural disasters. Recently established concepts in Information Technology and Geoinformatics allow for using sophisticated frameworks automatically generating database content, maps, or reports based on distributed static and real-time data. *Service-Oriented Architectures (SOAs)* and *Workflow Management* are two associated concepts in Information Technology to develop and distribute software in form of components within computing networks. With the approach of intercommunication between loosely coupled services that are accessed through well-defined interfaces, data sources and processing services from all around the world can be made available for disaster management in an efficient and powerful way.

In two use cases, an overview of current developments towards a SOA in the framework of the German initiative *Natural Disasters Networking Platform (NaDiNe*) is given. For the purpose of integrating real-time measurements into simulation models, the two use cases follow different technical approaches. Opportunities and restrictions of the implemented concepts in particular as well as of SOAs in general for supporting reliable disaster management systems will be discussed.

Introduction

Natural disasters like floods, earthquakes, tsunamis, storms or storm surges pose a major risk to human life and cause significant material damage. Not only long term trends show that the overall number of natural disasters is increasing, but there is also an increase in vulnerability to be stated (United Nations 2006). As a consequence, coordinated disaster management is getting more and more important and makes great efforts by many different parties necessary. Since coping with disasters is based on knowledge in many domains (i.e. environmental processes, social processes, planning etc.) it is a highly interdisciplinary task.

While data, expertise and technological components potentially usable for disaster management are spatially and institutionally distributed, in most cases systems supporting the management process are run locally on isolated servers of the responsible institutions. At the same time, further model components, data sources and expertise remain unused for disaster management, e.g. in research institutes. In order to make best use of the information available, however, new approaches are necessary for the collaboration of multiple parties.

With the upcoming of recent developments and standardization efforts in Information Technology and Geoinformatics, a technological and conceptual framework nowadays exists that can be used for enhancing the efficiency of disaster management significantly. Today sensors measure data from all over the world, high-performance computing clusters run simulations models and large network bandwidth allow for appropriate data transfer rates between the different locations. Besides these technical achievements, the existence of an additional conceptual framework is necessary for an operational information architecture to be realized. The paradigm of SOAs promises to provide the “glue” for making the single components work together, while keeping the whole architecture as flexible as necessary for efficient and reliable implementations.

In the context of the *Natural Disasters Networking Platform (NaDiNe*, Haubrock et al. 2006), an initiative of German research centers within the Helmholtz Association, developments towards transfer of research results into practice are currently realized. The overall aim is to provide information in the context of certain natural hazards and disaster events from the participating scientific institutions for users in disaster management and the public.

Within this paper, an overview of current developments towards Service-Oriented Architectures in the framework of NaDiNe is given. Hereby the term service will be defined in the beginning considering the context of its usage here. Afterwards, a short overview of relevant concepts and technologies related to Service-Orientation in Information Technology and Geoinformatics will be given. The following use cases show exemplarily some concepts and technological implementations of integrating scientific expertise into disaster management.

Services and Service-Oriented Architectures

What is a *service*?

The term service is widely used in the context of business and information technology. It is commonly understood as "a facility supplying some public demand"[1]. This implies that a service can be performed by different entities, i.e. a human being, an enterprise as well as a hardware or software device as long as the idea of a service consumer on the one and a service provider on the other side is valid (Fig. 1).

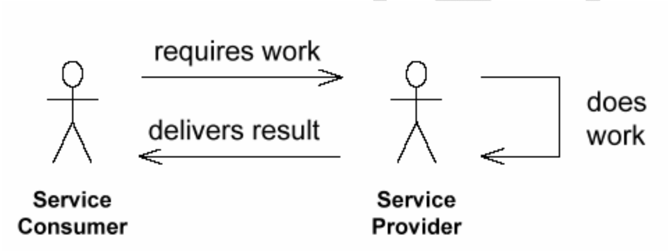

Fig. 1. *Relationship between service consumer and provider (Krafzig et al. 2004)*

While the original meaning of the term is rather general, the definition of a service in Information Technology is more restricted to computer programs providing services as executable code. For usage in the context of disaster management, where services need to be efficient and provide a "24/7 availability", a service is generally understood as a computer program that can be accessed by a service consumer (either a human or another computer/device). The term *web service* additionally requires a service to be platform-independent and remotely accessible. In the case of the

[1] Merriam Webster OnLine Dictionary: *http://www.m-w.com/dictionary/service*.

Service-Oriented Architectures established here, each service is also a web service.

A business process can be interpreted as a process the information system has to conduct for delivering a service to the user. Its aim is to create value by transforming an input into a more valuable output. Single activities within a business process can be performed automatically or by persons. In disaster management, a business process might be a task that an officer-on-duty has to perform, e.g. collecting relevant information or issuing a warning to decision makers. The second use cases discussed here shows how to make use of the SOA components using a business process management system.

SOA as a paradigm

The term SOA refers to service creation, interaction, presentation, and integration infrastructure capabilities (Nigam 2006) to build business-level software based on reusable components. While this definition originates from Information Technology, it refers to several different aspects and disciplines involved in this approach: business processes, design and usability. As a consequence, operational SOAs are in most cases interdisciplinary concepts involving the analysis of these aspects from different perspectives.

The new layer of abstraction introduced with service orientation now bares the potential to bridge the communication gap between IT developers and experts using these services, introducing a new business process-driven approach. For disaster management, this separation of concerns between definition and description of business processes and its actual implementation in form of computational services bares the potential to give consideration to the functional heterogeneity of partners involved. Public authorities or other institutions in disaster management are able to understand the semantics of the services and can therefore integrate them in their decision making process, while scientists and technicians provide and administrate these components.

SOA as a technology

The idea of SOA is in first hand a concept and as such not restricted to a certain technology. However, when talking about SOAs many people understand this concept as a specific implementation consisting of web services following platform-independent standards like XML, WSDL[2], UDDI[3] and SOAP[4]. The "SOAP-approach" of SOAs is probably the most-widely used and therefore has the advantage to be interoperable with many other services offered to build even larger systems. Another factor fostering the dominance of this technological approach is the existence of numerous tools and supporting frameworks to set up SOAP-based SOAs.

In Geoinformatics, it started in the late 1990's that the concept of monolithic Geoinformation Systems (GIS) has more and more been displaced by rather flexible distributed approaches, eventually leading to so-called Spatial Data Infrastructures (SDIs). SDIs achieve to provide access to GI services (business processes) in a structured way using certain XML-based specifications for description and data storing. SDIs can therefore be seen as realising the idea of SOAs as defined above. However, the *Open Geospatial Consortium (OGC)* as the leading standardisation initiative in this domain does not explicitly specify the usage of WSDL, UDDI and SOAP as base concepts. Instead, they specified web services to be accessed either directly via HTTP parameters (e.g. *WebMapServices(WMS)*) or using XML-based messages (e.g. *WebFeatureService (WFS)*). The standard data exchange format for spatial datasets is the XML-based *Geography Markup Language (GML)*, encoding vector datasets by using tags for geometries and attributes, respectively (Cox et al. 2002). With the current discussion of a *Sensor Web Enablement (SWE)* specification (Botts et al. 2006), a major step has been performed towards integration of real-time data into applications relevant for disaster management.

In order to make use of both, the OGC-specified components for easy-to-use mapping and data delivery services and the option to set up larger SOAs comprising more functionality, certain adaptations need to be implemented to connect both technical approaches.

Towards SOAs in natural disaster management

Connecting OGC web services for flood modelling

The existing standards of the OGC make the process of setting up a Service-Oriented Architecture consisting of data servers and mapping compo-

[2] Web Service Description Language
[3] Universal Description, Discovery and Integration
[4] originally: Simple Object Access Protocol

nents a straightforward task. The goal of the first use case is to set up a basic SOA for integrating real-time data into flood modelling and to make the whole process chain available to a certain remote user group in disaster management. The implementation is based on OGC standards and corresponding wrapper components to link additional services to the specified interfaces. The workflow is sketched in Fig. 1.

A user enters parameters into an HTML page and triggers a hydraulic model by submitting the form (1). This model is encapsulated as a web service. The parameters are read and incorporated in a request for spatial data needed for simulation. The request is sent in the next step (2) to the WFS. The instance of this service keeps synchronised with a data provider for real-time gauge measurements. In regular intervals, the WFS updates itself, so the newest data can be queried from the external service. The data format provided has to be transformed to an open data format that can be use by the services in the architecture, i.e. the hydraulic model as well as the WMS. This format is a specified schema of the standardised GML. With this approach, multiple data providers offering different data formats can potentially be integrated in the architecture, eventually being stored in a single standardised format.

The gauge data of the starting point is sent back to the model (3), which is now able to perform its simulation (4). The results of the simulation run are simulated water level datasets, which are uploaded to the WFS (5), so they will be accessible later on by the map. In step 6 the model sends a message back to the client about the success of the simulation together with a simulation ID that can be used to unambiguously identify the dataset. This ID is used to set up a query passed to the WMS (7), which in turn constructs a query for the appropriate dataset and sends a request to the WFS (8). After returning the data in GML format, the WMS generates a map out of the newly measured and simulated data and returns this map as an image back to the user, where it is visualized in the browser.

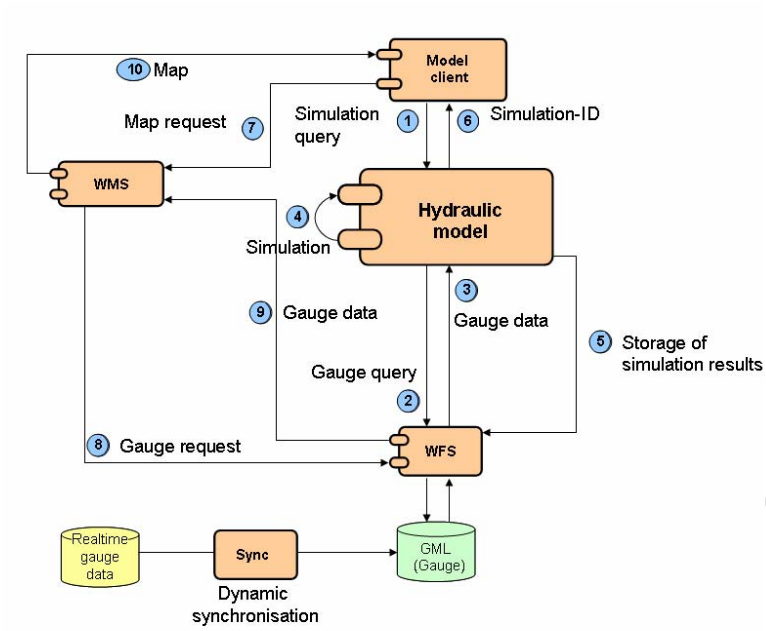

Fig. 2. *Service-Oriented Architecture of OGC web services and additional components*

This use case makes direct use of the interfaces specified by the OGC when accessing the web services. In order to integrate the simulation model into this SOA, the implementation of a new interface is necessary, since no standards exist from the OGC for generic web services to be used. The setup of the whole system is done manually, i.e. it is statically defined by hard-coding the process chain at design-time. The components of the model client and hydraulic model additionally have the tasks to generate WFS and WMS queries. This contradicts the idea of *separation of concerns*, which is fundamental for SOAs, i.e. that one single service has to provide only the functionality of a single business process.

The technical approach in the first use case is straightforward to implement, but misses an administrable component at runtime, keeping control of the overall process chain and triggering the single components one after the other instead of leaving the flow control to multiple components.

From OGC service architectures to business process management

For this reason, a more flexible approach is proposed here for a similar use case, based on SOAP web services and *Business Process Execution Lan-*

guage (BPEL) as the language for specifying the controlling instance. The overall aim of this implementation is the provision of digital maps that contain information about the results of simulation scenarios from a dyke breach model established at the GeoForschungsZentrum Potsdam. Instead of calling the services directly as in the first use case, a workflow is started with submitting the form eventually resulting in a map that shows potential dyke breach locations based on the input data and processed by the simulation model.

The technical part of the system consists of several components accessible via web services to provide the basis of a Service-Oriented Architecture. Two relevant components with respect to dynamic map creation in this use case are the dyke breach model itself (called *DykeBreachModel*) and a web mapping component (called *MapGeneratorService*) generating maps from both static base data and dynamic simulation results. The other service components comprise a *WorkflowEngine*, which coordinates the overall workflow, an *HTMLGenerator Service* and a *DataStorage Service* as sketched in Fig. 3.

The overall workflow of this scenario is triggered by the user who fills in the parameters, which are necessary to determine the location and to calibrate the model. With submitting the form, a message is sent to the workflow engine (1: *startWorkflow*) that coordinates the execution of the remaining process chain. The whole scenario is saved here as a business process. In step 2, the spatial input data is queried from the *DataStorageService*, returned to the workflow engine in an XML-based format and passed to the *DykeBreachModel* in step 4 with a call for *startSimulation*.

The model performs its simulation in a large number of iterations, which might actually induce further communication tasks between the model and the workflow engine. After the last iteration, the model service sends the result data to the workflow engine (5), which in turn inserts the data in the data storage repository (6).

With the data originating from the simulation, a new map can be generated by the *MapGeneratorService*. In step 8, the service is called, which in turn queries the *DataStorageService* for the appropriate data to be visualised in the map (9). The *MapGeneratorService* encapsulates a standardised WMS, while the *DataStorageService* relies on the functionality of a WFS.

The resulting map is saved as an image at the server and a message containing the image URL is sent back to the WorkflowEngine (11). The last processing step is performed by the HTMLGeneratorService, which creates a result HTML file with a reference to the newly created map image. This file is in a last step sent back to the user in step 14.

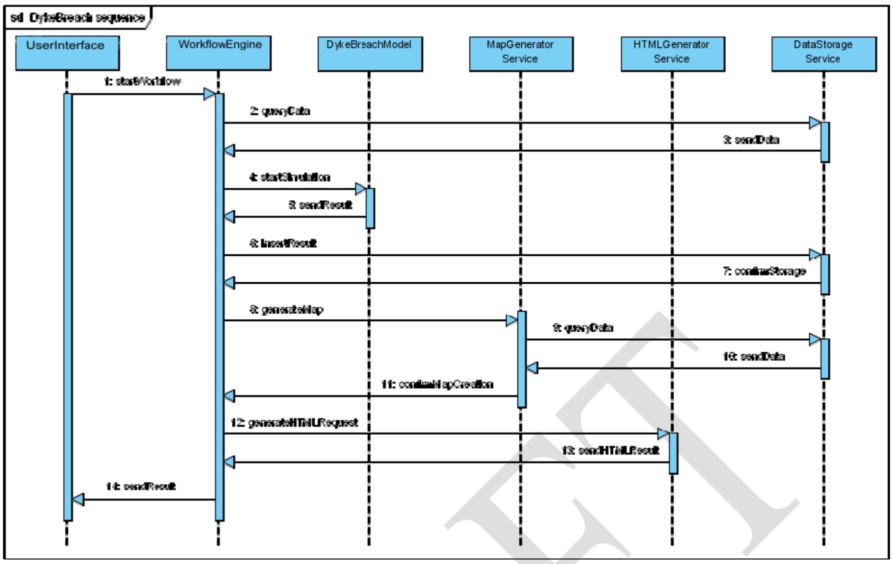

Fig. 3. *Simplified UML sequence diagram of typical workflow for dyke breach use case*

With this approach, the user accesses the workflow engine over a customized user-interface and waits for the result. Since the web services are implemented using SOAP and WSDL, they may be distributed over the Internet and at the same time be integrated into a workflow defined by a BPEL configuration file.

Processing-intense components can remain on appropriate servers while being accessed via SOAP messages over HTTP. The whole processing in the background remains transparent to the user and therefore makes the usage of the components a straightforward task. Standard OGC services like WMS and WFS can be integrated into the architecture by encapsulating them in SOAP-enabled wrappers.

Potential and limitations of SOAs in disaster management

Disasters like floods, earthquakes or storm surges are events where space and time are crucial. Thus, the acquisition, processing and analysis of geographic information are vital for disaster management. In the future, sensors of all kinds (from satellites, airplanes, in-situ measuring devices, cameras etc.) will generate masses of data to be used in business applications or information systems. The characteristics of future geographical data sets affecting performance can be summarized as follows. These data sets

- are large in size (several GB to TB),
- are distributed over many different sites,
- are time critical (real-time data),
- require massive processing power,
- need to be available in standardised formats.

As an important example, satellite images as one major source of near real-time spatial information are stored in raster files with up to hundreds of spectral bands. Today, the costly pre-processing steps of remote sensing images are performed before manually or semi-automatically. The main problem hereby is the fact that a lot of information is necessary in order to be able to perform these tasks. Knowledge of in-situ measurements, libraries providing typical spectral signatures, measurements of atmospheric data and geometrical locations are crucial for adequate processing of the image. Current research in this domain aims at facilitating the gathering and operational usage of these pieces of information in order to set up models, which are able to perform operational processing chains for remote sensing and other sensor data.

GI services provided in a SOA can be used as a key approach to compile the necessary information collected from sensors or simulation models for decision makers and to communicate between multiple parties. They are further usable for performing pre-processing steps in a more efficient way. However, human knowledge is in most cases required for these tasks, so concepts for efficiently integrating them into a SOA are needed.

Performance aspects of typical GI services

One specific characteristic of GI services is the manipulation of large datasets, which are increasingly distributed over different sites on the Internet. In order to setup a SOA based on real-time solutions serving and manipulating geographical data, performance aspects have to be particularly taken into account.

The formulation of requests in XML as realized in both approaches sketched here can be seen as advantageous in terms of self-description and security, but these benefits are achieved at the cost of bandwidth and performance. Data in an OGC-compliant SOA have to be encoded in GML, which tends to form a bottleneck that needs to be addressed for performance reasons. In order to reduce the size of messages and data exchanged in SOAs, the only option is to represent data binary, either in plain - or even better - compressed format. One approach to establish a concept for realizing binary representations of XML data is proposed by the *Fast Web Services* initiative. It aims at “enabling fast and efficient end-to-end, interoperable, Web Services for a whole spectrum of devices, from mobile phones to large back end servers” (Sandoz et al. 2003).

One possibility is to use the technology of *Abstract Syntax Notation number One (ASN.1)* (Dubuisson 2001) established in telecommunications for encoding data in a way such that it can eventually be re-converted into XML. By using the *Packed Encoding Rules (PER)* as compact binary representation of ASN.1-encoded data, compression factor of >20 are possible. The major drawbacks of this concept are security constraints and en-/decoding overhead. While former problem is targeted at with the aid of signatures and encryption, the latter issue could be accepted as the less significant performance bottleneck compared to plain GML encoding.

Since conventional databases are not able to cope with the resulting performance issues when storing massive amounts of datasets, standard database technologies will fail for these applications. A first solution could be to incorporate data warehouses for distributed storage of these data. Data warehousing is able to manage large amounts of historic and actual data. However, the relatively long access time to the data remains an issue for applications in disaster management. Since storing data on disk is the main reason for the relatively long access times (e.g. 10ms), one solution could be to store data of the warehouse in main memory (Graves 2002). The access time can be reduced to ~1ms when data sets are stored in their native format. Following this idea, even whole satellite maps of hundreds of GB might be accessed in an acceptable time frame.

The remaining aspect of processing power is suitably tackled by setting up massive parallel systems of computers. The overhead of administering the distribution in processing power is significantly outweighed by the gain of performance by this kind of clusters. One major advantage of this approach is the flexible scalability of these systems and the possibility to keep processing at the site where the data are collected (sensor or modeling site). The resulting architecture summarizing the key ideas to improve performance of SOAs for disaster management is sketched in Fig. 4.

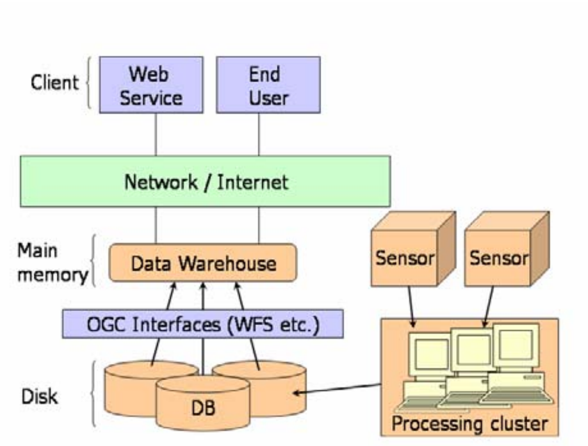

Fig. 4. *Architecture of data storage and processing distribution for large real-time datasets in a SOA*

Conclusion and Outlook

Service-Oriented Architectures have been established in many business domains as they keep the promise to significantly enhance efficiency in performing business processes. Consequently, they bare the potential to enable a decision maker in disaster management to access data at runtime, run simulations without being aware of the actual implementation, and generate maps or reports for information aggregation. Scientists as well as practitioners can generate, analyse and compare results of different scenarios in a distributed user friendly fashion.

The availability of sensor data from all around the world is steadily growing. While a variety of different data and information sources exist for disaster management, this potential hasn't been exploited yet although the technical and conceptual frameworks exist. Besides current developments in the context of the OGC, further technical and conceptual approaches are available from other disciplines, like ASN.1 from the telecommunications sector. Combining these parts allow for setting up efficient SOAs to be used in operational workflows.

Combining the possibilities of Information Technology with the knowledge of the people is one of the main benefits and challenges of the SOA approach. Although technical and conceptual problems remain to be solved, the major obstacles to be overcome are in many cases rather organizational, legal and political.

References

Botts M, Percivall G, Reed C, Davidson J (2006) OGC® Sensor Web Enablement: Overview And High Level Architecture. (White paper)

Cox S, Daisey P, Lake R, Portele C, Whiteside A. (eds.) (2002) OpenGIS® Geography Markup Language (GML) Encoding Specification.

Dubuisson O (2000) ASN.1 - Communication between heterogeneous systems. Morgan Kaufmann Publishers, San Francisco.

Graves S (2002) In-Memory Database Systems. Linux Journal, 9/2002.

Haubrock S, Dransch D, Kreibich H, Merz B, Plattner S, Voigt S, Mehl, H (2006) Natural Disasters Networking Platform (NaDiNe). In: Amman W, Haig J, Huovinen C, Stocker, M (eds) Proceedings of the International Disaster Reduction Conference, vol 1, Swiss Federal Research Institute WSL, Davos, p 74

Krafzig D, Banke K, Slama D (2004) Enterprise SOA: Service-Oriented Architecture Best Practices. *Prentice Hall, New Jersey*.

Nigam S (2006) Service Oriented Development of Applications (SODA) in Sybase Workspace. (White paper)

Sandoz P, Pericas-Geertsen S, Kawaguchi K, Hadley M, Pelegri-Llopart, E (2003) Fast Web Services. Sun Developer Network, URL: http://java.sun.com/developer/technicalArticles/WebServices/fastWS/

United Nations (2006) Implementation of the International Strategy for Disaster Reduction. (Sustainable development: International Strategy for Disaster Reduction - Report of the Secretary General)