Interoperability of Simulation and Geoinformation Standards

Ingo Simonis, Andreas Wytzisk
Institute for Geoinformatics, University of Muenster
Robert-Koch-Strasse 26-28
48149 Muenster, Germany
+49-251-83-30057, +49-251-83-39763
{simonis, wytzisk}@ifgi.uni-muenster.de

Dr. Ulrich Raape
Fraunhofer Institute for Factory Operation and Automation IFF
Sandtorstrasse 22
39106 Magdeburg, Germany
+49-391-4090-359, +49-391-4090-93-359
raape@iff.fhg.de

Dr. habil. Thomas Schulze
University of Magdeburg
Universitaetsplatz 2
39106 Magdeburg, Germany
+49-391-6712825
tom@iti.cs.uni-magdeburg.de

Keywords:
HLA, OpenGIS, OGC Specifications, Interoperability, Integration, Simulation

ABSTRACT: Like the High Level Architecture (HLA) in the simulation domain, mature interoperability standards and specifications have evolved in the spatial domain. Here, recent development mainly focuses on the interoperability of data processing capabilities and geographic data itself. Both are made available to the World Wide Web by standardizing web interfaces. Interoperability at the data level is achieved by defining the Geography Markup Language (GML) as an XML encoding for the transport and storage of geographic information, including both the spatial and non-spatial properties of geographic features.

As there are a couple of specifications available describing the interchange and use of static geographic data, the demand to reproduce and access dynamic operations via the web is growing. Within the spatial domain, first actions have been taken by standardizing interfaces for the “Sensor Web Enablement” (SWE). Being aware of the fact that a simulator does not differ from a sensor regarding the provision of spatio-temporal data (it only differs in the way how it estimates the requested value and its virtually temporal independence), we had an obvious starting point to integrate dynamic capabilities (simulation) into the web service world. Simultaneously, live data measured by real sensors will be made available to running HLA federations in an interoperable way.

1. Introduction

In the late eighties, the call arose for linking up geographic information software with simulation engines (Nyerges 1993). Since that time, GI-software has evolved from an entirely desktop driven approach to an internet based infrastructural design. Simultaneously within the simulation domain, the development shows a rising level of distribution of individual simulation components. Both have in common to achieve a high level of interoperability and reusability – unfortunately only within their particular domain.

1.1 Geoprocessing Domain

Spatial data infrastructures (SDI) are currently one of the hot topics within the geographic information domain. Mature interoperability standards and specifications have
evolved in the last years, mainly driven by the work of the OpenGIS Consortium (OGC). OGC is an international industry consortium of more than 230 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications. OpenGIS® Specifications support interoperable solutions that “geo-enable” the Web. It represents one of the largest organized forces in the global geo-processing marketplace. Its objective – to establish a global geoinformation infrastructure – is achieved by developing unique service specifications. These specifications serve as a generic programming interface and allow the interoperability of different GIS components (Buehler and McKee 1998). Within the scope of the current OGC Web Services 1.2 Testbed initiative, new interfaces have been developed that allow the access to live sensors, including basic control and management potentials. These interfaces have not specification status yet, but represent very promising approaches. Based on the fact that a simulator does not differ from a sensor regarding the provision of spatio-temporal data (it only differs in the way it estimates the requested value and its virtually temporal independence), these interfaces where extended to fulfill the specific needs of simulators to be made accessible via the World Wide Web.

1.2 Simulation Domain

Regarding the integration of distributed heterogeneous simulation models are the specifications of the High Level Architecture (HLA) still state of the art (IEEE 2000). The HLA framework, born in the military domain in 1996, has broken into the civil domain during the last five years and constitutes the only standardized simulation framework so far (as an advancement of (DMSO 1998)).

Still, there are no simulation frameworks standardized that are based on web services. Though first approaches and prototypes exist regarding a simulation service providing (Wiedemann 2001; Gebert and Osterburg 2002), the term “Web-based Simulation” is still dominated by the idea of interoperable simulation components, which are distributed over the internet and integrated by specific integration software. This software facilitates the download of the needed components to a single machine and executes them locally (Page, Buss et al. 2000).

2. Integration – High Level

Due to their complementary capabilities as well as the international standardization, HLA and OGC Web Services form a suitable basis to specify simulation services. Those services, shaped as interoperable GI-services that are able to become integrated into spatial data infrastructures, make it possible to control simulation models within the World Wide Web and make the simulation results accessible easily. To use the maximal performance of both – HLA and Web Services – the design of an integrating architecture becomes necessary to arrange the underlying fundamental different software concepts. The main task is to forge links between the stateful, closely coupled HLA world and the stateless OGC world. The following points have to be addressed:

- To control simulation experiments and to access the simulation output, corresponding OGC Web Service interfaces have to be created.
- HLA federations have to be made controllable from external processes. Whereas OGC Web Services exist permanently, federates live only during a simulation run. To allow the ad hoc creation of distributed simulations, federates have to be made permanently available and controllable as part of a federation, using suitable management services. A possible approach is discussed in detail in (Wytzisk, Simonis et al. 2003).
- Geographic objects have to be made interchangeable within HLA federations. To ensure a highly efficient information exchange with OGC compliant services, the specification of a reference FOM becomes necessary. You have got to be aware that this FOM follows the Open Geo-data Model of the OGC (Buehler and McKee 1998) as closely as possible.
- The concept shall enable the integration and use of OGC Web Services within HLA federations. To ensure HLA compliancy, specialised federates have to be created that facades selected OGC Web Services. Those federates will use the declaration and object management instructions defined within the HLA standard.

To ensure a high level of interoperability, the newly defined software concept has to avoid any proprietary extensions of any of the standards mentioned. Figure 1 illustrates this concept.
3. Sensor Web Enablement – SWE

The Sensor Web Enablement constitutes the OGC approach to make sensor data available via web services. It was initially designed to fulfill the following needs:

- Describe sensors in a standardized way
- Standardize the access to observed data
- Standardize the process of what is commonly known as sensor planning, but in fact consisting of the different stages planning, scheduling, tasking, collection, and processing
- Building a framework and encoding for measurements and observations

The demands expressed in these points had led to the development of four different specifications which will be described in detail. To describe sensors, the markup language SensorML had been created. The service interface specification of a Sensor Collection Service provides access to observed data whereas the Sensor Planning Service provides an interface to organize the observation of natural phenomenon. The results of these observations are XML encoded following the standard described in Observation & Measurement.

During the development of the specifications mentioned above, it became obvious that sticking to the synchronous character of OGC web services was not possible any more. The execution of complex observations (like ordering satellite images from a certain location with maximal cloud coverage of five per cent) or complex simulation runs may take virtually any time to fulfil. That’s why the SPS is one of two OGC web services that are not stateless any more. The other stateful service is the Web Notification Service. Designed by the authors (Simonis and Wytzisk 2003), this messaging service fulfills the requirements of a complex communication service by providing the communication basis for asynchronous service handling.

Sensor Collection Service

The objective of OGC’s SWE activities is to develop a set of services that allows to discover, find and bind access functionality (see also figure 2) to a single or constellation of real-time sensors heretofore referred to as a Sensor Collection Service (SCS)(McCarty 2002). Once accessed, a SWE client application receives readings from a sensor via the SCS where it displays or processes the data in a variety of manners.

Sensor Planning Service

The Sensor Planning Service (SPS) is intended to provide a standard interface to collection assets (i.e., sensors, and other information gathering assets) and to support systems that surrounds them (Lansing 2002). A SPS not only has to support different kinds of assets with differing capabilities, but also different kinds of request processing systems, which may or may not provide access to the different stages of planning, scheduling, tasking, collection, processing, archiving, and distribution of resulting observation data. The SPS is designed to be flexible enough to handle such a wide variety of configurations.

Web Notification Service

The WNS is an asynchronous and stateful service. It is a web interface (e.g. operated by the SPS) that allows sending notifications to a client with well structured content. To enable any kind of dialogue between the user and an invoking service, functionality is provided that enables the user to answer asynchronously with any kind of structured content. Currently four different notification delivery channels are supported: email, SMS, http POST, and instant messages. It had been avoided to specify a “SWE-WNS” rather than a WNS, because web notification functionality will become necessary for other web services as well at the moment SDI usage will become real business (Simonis and Wytzisk 2003).

SensorML

SensorML is a conceptual model that describes the geometric, dynamic, and observational characteristics of a sensor using XML encoding. Sensors are devices for the measurement of physical quantities. There are a great variety of sensor types from simple visual thermometers to complex earth observing satellites (Botts 2002).

Observation and Measurement

The Observations and Measurements Engineering Specification describes a conceptual model that allows the de-
piction of spatial and temporal variant readings which values are the results of the estimation of some natural phenomenon. The representation is based on the specified XML encoding. For Cox, measurement usually refers to the measuring device and procedure used to determine the value, such as a sensor or observer, analytical procedures, simulations or other numerical processes. The carrying out of the procedure to estimate the value of the phenomenon is called observation (Cox 2002).

3.1 Different Scenarios

Having a closer look on the integration process of simulations into web services, it becomes obvious that there are two different scenarios to be distinguished. In the first case, the simulator is continuously running, independently of the user. It is started, parameterized and maintained by the simulation provider and has no interface to the user other than necessary to enable access to the simulation results. These simulators provide e.g. temperature-, precipitation-, or speed of wind-forecasts. It is even possible that a user will not notice at all that the provided data is resulted from calculations rather than real measurements (e.g. if the values are interpolated due to a rather wide-ranging measuring network). Those simulators will be encapsulated to a simple datastore that is accessible using a Sensor Collection Service. It can be handled purely synchronous and follows the service trading (publish – subscribe – bind) paradigm.

In the second case, a simple encapsulation is insufficient. All observations or just simulations that require preceding feasibility studies, complex control and management activities, or intermediate and/or subsequent user notifications, are not handleable synchronously anymore but become heavily asynchronous. In this second case the service interactions become much more complex. A list of services will become necessary. The main part regarding simulation management plays the Sensor Planning Service (besides Web Notification Service to receive notifications about finished simulation steps, and Sensor Collection Service to access the simulated data eventually). The sensor planning services provides interfaces that allow requesting a simulation run, feeding the necessary parameters to the simulator and to start the simulation run. Figure 2 illustrates the typical sequence of interactions occurring to start and maintain a simulation run using a SPS:

Sensor Planning Services as facades for simulation models are the first step within the integration process of simulators into web services. Currently, a rather tight coupling is unavoidable. This fact will be improved when the first generic simulation interfaces become available which support simulation management tasks independently of the underlying simulation system.

![Figure 2: Interaction Sequence of a simulation run using a SPS (Wytzisk 2003)](image-url)
3.2 High Level Architecture

To attain availability of simulation models using a Sensor Planning Service, it becomes necessary to specify a HLA based framework that allows the management of any distributed simulation run from any process, independently of the kind of federates of federations. This management functionality is not part of the standard IEEE 1516: Status and behavior of a federate are not defined after it has once joined or left a federation. There are basic management capacities available only in case a simulator already exists as a physical process and a federate has already joined. But even these management capacities are not usable from external processes like web services but are only to the federate’s disposal.

To make HLA based simulation available via WWW, means to control the simulation run and to access the results, we will show a concept that will allow the following points without infringing the IEEE 1516 specification (Wytzisk 2003):

• To initiate federations externally
• To control start and termination of federates using an external management process
• To make the simulation results available for external, non HLA-processes

On the one hand, the enhancements are based on the concept of a management federation that allows instantiation and management of the actual simulation federation and corresponding federates. “Simulation Federation” defines the federation which actually executes the simulation experiment. On the other hand, they are based on a specified reference FOM.

A management federation consists of a bulk of federates. This federates belong to one of the four types:

• Management federates
• Management control federates
• Management bridge federates
• Management interaction federates

Management Federates are responsible to start local or remote simulation federates. The simulation federates are described by federate descriptors that are published within the management federation. The object class definitions are part of the Management FOM, which itself is independent of the simulation domain. Management Control Federates send control commands as interactions into the federation – e.g. to start or terminate simulation federates and simulation federations – which are received and processed by the management federates. Syntax and semantic of the control commands are defined in the FOM as well. Usually, management federates and simulation federates exist as distinct processes and are not able to communicate except of using proprietary channels. To be able to control a simulation federation, we will have to extend it by a specific management interaction federate that is closely coupled to a specialized management federate, the so called management bridge federate. This management interaction federate sends commands to the simulation federates, e.g. for termination purposes. All control and callback interactions that are subscribed by the management interaction federate are described in a simulation interaction FOM that has to be supported by all simulation federates that will participate at the described practice. Figure 3 shows the different federate types in UML notation.

Figure 3 Federate types in UML notation

As a result of the close coupling with the management bridge federate management interaction federates allow to:

• Retrieve information about the members of a simulation federation from the management federation and therefore make this information available to external processes.
• Send user-specific interactions into the simulation federation. Hence it becomes possible to control a simulation federation from external processes by make use of the “management control federate” – “management bridge federate” bridge.

4. Integration – Low Level

The represented concept should be demonstrated at a hydrological use case shortly. In the case of flooding, the usability of hydrological and meteorological observation and prediction data plays a decisive role for the success of high water resistance actions. The actions where organ-
ized by the crisis management group based on actual and predicted precipitation and run off data.

The implemented prototype facilitates the access to
- The current precipitation data
- Current water levels and
- Predicted run offs.

The precipitation and water level information is provided by a SCS compliant interface. The prediction of the run off is realized using a HLA compliant simulation model. The simulator is parametrised and executed using a SPS with underlying management federation. Once started, the simulator provides run off predictions based on the current precipitation. The crisis squad will be informed about the availability of new data – which is accessible using a SCS – by a WNS. Figure 4 demonstrates the order of events.

The simulation model is based on the SCS Curve-Number method, developed by the U.S. Soil Conservation Service (Maidment 1993), (USDA Soil Conservation Service 1972), (Dingman 1994).

5. Conclusions

The shown concept allows the realization of an information system – which is entirely based on international standards – that allows the interoperable integration of simulation models into a spatial data infrastructure, built up by web services. Simulation services which are based on the High Level Architecture will be integrated into the OpenGIS-based service infrastructure. The SWE services, which are independent of a concrete user domain, are sufficiently flexible to supply specific simulation models in an interoperable way.

On the simulation side facilitates the represented HLA framework with management and simulation federations the integration of simulation models as well as the use of spatial data infrastructural features within the simulation model itself.

On a conceptual level, it has to be discussed how to realize a simulation environment access layer that is independent of the High Level Architecture. This layer would be placed between the SPS and the federation controller service, as shown in figure 5.
Coupled with a standardized description and parameterization concept for simulators, this layer would allow the complete generic implementation of SPS and hence lead to a higher level of interoperability eventually.

References


Author Biographies

INGO SIMONIS is a project manager at the Institute of Geoinformatics of the University of Münster, Germany. He holds a master degree in Ecology and currently works on his dissertation, issuing the usability of simulation results within a web service environment and the subsequent integration into a web based decision support framework. His further research focuses on the integration of service-based geographic information infrastructures and simulation-based software architectures as well as mobile computing concepts, architectures, and techniques.

ANDREAS WYTZISK is a project manager at the Institute of Geoinformatics of the University of Münster, Germany. He studied Physics and Geography in Heidelberg and Münster, Germany co-founded the pro_Plant company for agricultural and environmental informatics in 1996, and finished his Ph.D. thesis in the beginning of 2003. His research topics include the integration of service-based geographic information infrastructures and simulation-based software architectures as well as logistical problems in agriculture and forestry.

ULRICH KLEIN is a Project Manager at the Department of Environmental Engineering of the Fraunhofer Institute IFF.
He received his Ph.D. from the Otto von Guericke University of Magdeburg, Germany. He holds a masters degree in Industrial Engineering from the University of Karlsruhe and has two years experience as a Project Manager for Command, Control and Communication Systems for Public Safety and Security in Europe. His research topics include Emergency Management, Logistics, Geographic Information Systems and HLA-based Software Architectures.

**THOMAS SCHULZE** is an Associate Professor in the School of Computer Science at the Otto-von-Guericke-University, Magdeburg, Germany. He received the Ph.D. degree in civil engineering in 1979 and his habil. degree for computer science in 1991 from the University of Magdeburg. His research interests include modeling methodology, public systems modeling, manufacturing simulation, and distributed simulation with HLA. He is an active member in the ASIM, the German organization of simulation.