Open Service-oriented Architecture for Environmental Risk Management Applications

Thomas Usländer, Fraunhofer IITB, Germany
Ralf Denzer, Environmental Informatics Group, Germany
Reiner Güttler, Environmental Informatics Group, Germany

Abstract

Risk management during disasters consists of the set of preventative, integrated actions taken to deal with risk identification, analysis, and mitigation. Information Technology (IT) plays a key role in this. However, the ability to share all relevant data, especially in disasters which cross international borders, is often very limited because risk management tasks are mainly handled by public institutions on a variety of administrative levels, each with their own IT systems. The integrated EU project ORCHESTRA (Open Architecture and Spatial Data Infrastructure for Risk Management, http://www.eu-orchestra.org) has taken up this IT challenge. ORCHESTRA specifies, implements and demonstrates an open, service-oriented architecture to improve the syntactic and semantic interoperability of risk management applications. The architectural approach is specified in the Reference Model for the ORCHESTRA Architecture (RM-OA). The architecture design follows the ISO/IEC 10746 Reference Model for Open Distributed Processing and adapts it to service-oriented architectures.

This paper illustrates the vision, the major IT challenges and the architectural elements of the RM-OA. It motivates the architectural design decisions from the perspective of the requirements triangle “user requirements”, “system requirements” and “compliance to existing and emerging standards”. The paper also provides an overview about the specification and the specification process of the information and service viewpoints including their rules to map to a chosen service platform, here W3C Web Services. The paper concludes with an outlook about the future role of semantic technologies in such geospatial architectures.
1 Vision and Challenges

The vision of the ORCHESTRA project is to contribute to future “ideal” IT support of environmental risk management. Such ideal support would increase information and service quality for end users, make information available on demand, whenever and wherever needed, in a personalised way, and would enable information and service providers to provide high-quality services at considerably lower cost, and when necessary nearly on-the-fly. One part of this ideal support would be an IT infrastructure which provides seamless access to resources (information and services) across organisational, technical, cultural and political borders, thus overcoming real-world heterogeneity which at the same time can be responsive to future requirements. Whether this IT infrastructure is in itself composed of several approaches is not relevant, as long as amongst themselves they are inter-operable. The final goal is plug-and-play of environmental risk management resources, services and applications, providing end users with cross-border services for risk and disaster management which they lack today. There are many obstacles which have prevented the realisation of the vision described above in the past. (Denzer 2005) carried out a discussion of several fundamental challenges based on a simple systems-of-systems model. They may be classified into the four following categories:

- Syntactic Inter-Operability: how to overcome technical heterogeneity of all sorts
- Semantic Inter-Operability: how to overcome ambiguities and different interpretations
- Organisational Context: how to overcome cross-border situations of all kinds
- Generics: how to deliver sustainable and re-usable concepts and components independent of the application domain, the technological infrastructure and the organisational context.

In the end the question remains how to build an IT infrastructure which is capable of facing these challenges. There are the following architectural frameworks and standardisation organisations that provide partial but not complete solutions to the above challenges:

a) OGC\(^1\) is highly relevant as most of the risk management information and services have spatial properties.
b) W3C\(^2\) is highly relevant, as it provides standards of commonly used middleware environments such as the W3C Web services.
c) OASIS\(^3\) is relevant, as it provides standards in general for the application of Web services and in particular in the context of user and rights management.
d) Semantic extensions, currently investigated in the context of W3C and OASIS, are relevant as they provide concepts and potential future standards in the area of semantics.

This article describes how the major elements of these frameworks may play together in order to tackle the IT challenges described above. This is the essence of the ORCHESTRA approach.

\(^1\) Open Geospatial Consortium, http://www.opengeospatial.org
\(^3\) Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org
2 The ORCHESTRA Approach

2.1 What is the ORCHESTRA Reference Model?

The ORCHESTRA Reference Model for geospatial service networks is built upon two main pillars: a process model and a conceptual model.

The ORCHESTRA process model follows an incremental, iterative approach for the analysis and design phases. Usually, a multi-step breakdown process across several abstraction layers is necessary to analyse user requirements and map them to the capabilities of a service platform. ORCHESTRA distinguishes between an abstract service platform that is specified independently of a given middleware technology and a concrete service platform (see Figure 1):

- The abstract design phase leads to platform-neutral specification following the rules of the abstract service platform provided by the ORCHESTRA Reference Model. They represent the functional requirements (→ abstract service specification), informational requirements (→ information model) and non-functional requirements (→ specification of the quality of service (QoS)) of the problem domain.
• The concrete design phase maps the abstract specifications to a chosen concrete service platform. In the current ORCHESTRA project this is the ORCHESTRA Web Services platform consisting of the rules of the W3C Web services and a profile of the Geography Mark-up Language (GML) as the current mainstream service platform technologies for geospatial applications.

• In the engineering phase the platform-specific components are organised into service networks taking into account the QoS requirements and translating them into operational policies.

In practice, these individual phases are often interlinked and repeated in an iterative fashion. Sometimes the abstract design phase is not required in the first place. Furthermore, existing service and OGC service standards for Web services make a pure top-down approach improper. Thus, in practice, a middle-out design approach is often the appropriate method.

However, what is required is a clear structure for the documentation of the ideas and the results of the design phases. As OGC, ORCHESTRA has adopted ISO/IEC 10746, Reference Model for Open Distributed Processing (RM-ODP), for this task but adapted it to the design of geospatial service networks (Usländer (Ed.) 2007).

The ORCHESTRA Architecture (OA) significantly help in this design phase as it provides a generic modelling toolbox in terms of pre-defined but generic information and service types (OA services) upon which the functional and informational user requirements may be mapped. It is specified itself as an abstract architecture. Of course, not all requirements may be directly mapped to existing generic information and service models. Thus, the ORCHESTRA Architecture also comprises a conceptual model\(^4\) that provides detailed rules about how to specify in UML an information model\(^5\) and service model (additional interface and service types) that fit to the pre-defined ones and adapt them for a particular application. Such additions lead to ORCHESTRA Application Architectures tailored to satisfy dedicated thematic user requirements which are expressed in thematic information models and thematic services.

However, ORCHESTRA does not stop at the abstract level but also provides an ORCHESTRA Implementation Architecture for the ORCHESTRA Web Services platform. Here, ORCHESTRA delivers a software toolbox comprising implementation specifications and implementation components derived from and compliant with the abstract specifications. For the thematic information and service models of an application architecture, tools are provided to map them to this platform.

2.2 What has ORCHESTRA to offer?

The promise of the ORCHESTRA project is manifold and is targeted at both designers and software engineers of a geospatial SOA. It encompasses the following elements:

\(^4\) In modelling terms, this conceptual model is a meta-model.
\(^5\) In modelling terms also referred to as application schema consisting of feature types and their relationships.
• A process model compliant with an ISO standard (RM-ODP) and tailored to the design and engineering of geospatial SOAs.
• A reference model for the design of geospatial SOAs.
• An open abstract architecture containing design rules for information and service models.
• Specifications, on both the abstract level and specific to the W3C Web services platform, of the most important generic architecture services derived from but not restricted to the needs of environmental risk management applications.
• Software engineering components, mostly offered under an open source license, for the development of service networks including
  - a Java-based software framework called OSCF (ORCHESTRA Service Container Framework) that comprises APIs for common service functionality (e.g. the service capabilities, access control) and therefore facilitates the implementation of additional services according to the ORCHESTRA approach,
  - implementation of indispensable architecture services integrated into the OSCF
  - adapters to industry standard services (e.g. the OGC Catalogue service with both the eBIM and the ISO Application Profile),
  - design support for the mapping of service and information models from the abstract level (UML) to the Web services platform (WSDL, XML/GML),
  - a Java-based software framework for the integration of source systems (e.g. for relational databases) into an ORCHESTRA service network, and
  - utility applications (e.g. for user management, service monitoring, catalogue navigation).
• Specification and implementation of risk management services based on clear end-user requirements and validated in real-world pilot applications.

2.3 What are the benefits of using ORCHESTRA?

The trend towards SOAs for the set-up and deployment of distributed environmental risk management systems is not only a technical discussion about the best middleware for such systems. Above all, it is a technological answer to the user request for efficient and effective cross-domain information and service integration such that answers to complex risk management questions and situations may be obtained more efficiently and with higher quality. And here is the key to where the benefits of the ORCHESTRA approach may be found. Such systems-of-systems require long-term thinking and sustainable design abstracting from but trading on technological advancements of middleware. Designers and modelers of such distributed system architectures are those who may benefit from the ORCHESTRA approach in the first place. The amount of well-specified functionality and information models both on the generic architectural level and on the risk management level may dramatically reduce the effort when breaking down user requirements into re-usable components. ORCHESTRA gives to the designer the foundation and important building blocks for the architecture. Furthermore, as elements of ORCHESTRA architecture are being fed into the standardization process at OGC and strategic European initiatives such as INSPIRE and GMES, the resulting system architecture is already inline with the future developments.
Nevertheless, ORCHESTRA also provides immediate support for software developers who are about to design and implement geospatial applications and service-oriented architectures today, possibly directly for the Web services platform with the need to integrate existing OGC-compliant services. In this case, the existing software development tools and frameworks, as listed above and already validated in pilot applications, as well as the series of ORCHESTRA implementation specifications and service components may immediately be used. This boosts software productivity, reduces the testing effort and increases the level of interoperability.

3 Elements of the ORCHESTRA Architecture

The ORCHESTRA Architecture has to face the problem of integrating environmental risk management systems that are networked across and between organisations. This is the objective of the ORCHESTRA Service Networks as running instances of the ORCHESTRA Architecture.

The operational components of an ORCHESTRA Service Network are the ORCHESTRA Service Instances (OSIs). OSIs offer their functionality and interact among each other according to the ORCHESTRA protocol, i.e. the set of the ORCHESTRA rules given by the ORCHESTRA Meta-model (OMM) as described below. By their joint functionality and interaction, they resolve the gap between the information demand of the user and the existing resources (data and services) offered by source systems. OSIs are organised in the following functional domains (see Figure 2):

![Figure 2: Functional Domains in an ORCHESTRA Service Network](image-url)
Software components in the **User Domain** provide the interface to a user component (a human or another software component). Their interaction among themselves is outside the scope of an OSN, i.e. they may use a native protocol. However, when interacting with an OSI, they have to use the ORCHESTRA protocol.

OSIs in the **Mediation and Processing Domain** provide the main functional part of an OSN. They mediate the service calls from the User to the Integration Domain based on meta-information exchanged with the components of the Integration Domain (e.g. by means of a publishing or a retrieval pattern).

OSIs in the **Integration Domain** provide support for the integration of source systems into an OSN. The OSIs in this domain have two-sided interfaces. On the one hand, they interact with other OSIs according to the ORCHESTRA protocol. On the other hand, they interact with the components of the Source System Domain according to their native protocol.

The **Source System Domain** incorporates the so-called source systems, i.e. the systems and system components (e.g. a relational data base) of a thematic application area (e.g. risk management) to be integrated into an OSN. In practice, this means that their data and functionality have to be wrapped with an ORCHESTRA-compliant service interface. In order to facilitate this re-engineering process, ORCHESTRA provides a dedicated software framework (see above).

### 3.1 Abstract Service Platform

On the level of the abstract service platform, the ORCHESTRA Architecture provides the following elements:

- A description framework and document templates for the textual specification of interface and service types.
- A coherent set of rules to specify interface, service and feature types in UML and to organise them in service and information models. This rule set is referred to as ORCHESTRA Meta-model (OMM). Key aspects of the OMM are:
  - It is an extension of the General Feature Model (GFM) used in the OGC Reference Model. The OMM treats both information and service aspects in a consistent manner.
  - In contrast to the mandatory usage of ISO 19115/19119 in the GFM, the OMM information part does not prescribe a particular meta-information model but just provides rules about how to specify meta-information models. This approach leads to a higher flexibility as meta-information in the OMM is considered to be purpose-specific, e.g. for the purpose of discovery a different set of meta-information elements may be defined than for service composition (Schimak et al 2007).
  - The OMM service part puts the interface type into the spotlight for re-usability. Interface types are specified such they may be re-used across several service type specifications. Examples of the application of this concept include the service capabilities interface type that is mandatory for all ORCHESTRA service types or the schema mapping interface type that is being re-used in a variant of the ORCHESTRA Feature Access Service.
<table>
<thead>
<tr>
<th>Service Type Name</th>
<th>Overview Description</th>
</tr>
</thead>
</table>
| OA Basic Service  | Interface types enabling a common architectural approach for all ORCHESTRA Services:  
• self-description of service instances (capabilities)  
• synchronous and asynchronous interactions  
• transactional support  
Furthermore:  
• predefined exception types |
| Authentication Service | Proves the genuineness of principals (i.e. the identity of a subject which may be a user or a software component) using a set of given credentials. Selected authentication mechanism is up to implementation specification. |
| Authorisation Service | Provides an authorisation decision for a given authorisation context. Selected authorisation paradigm is up to implementation specification. |
| Catalogue Service | Ability to publish, query and retrieve descriptive information (meta-information) for resources (i.e. data and services) of any type.  
• not tied to a particular schema of a meta-information standard (e.g. ISO 19115)  
• supports application schemas for meta-information that are designed according to the ORCHESTRA rules  
• May be used as a data catalogue, service registry or both.  
• May be based on other standard catalogues like OGC Catalogue or UDDI. |
| Document Access Service | Supports access to documents of any type (textual documents, images). A document is considered to be a specific kind of a feature type. |
| Feature Access Service | Selection, creation, update and deletion of feature instances and feature types available in a service network. Features provided are instances of a certain feature type defined in an ORCHESTRA Application Schema. Interface may be re-used by more specific access services using interface inheritance. |
| Map and Diagram Service | Visualizes, symbolizes and enables geographic clients to interactively visualise geographic and statistical data.  
Transforms geographic data (vector or raster) and/or numerical tabular data into a graphical representation using symbolization rules. The main output of this service is an image document which may be a map, a diagram or a thematic map (visualization of the spatial distribution of one or more statistical data themes). |
| Name Service | Encapsulates the implemented naming policy for service instances in a service network, e.g. creates globally unique service instance names using a defined naming policy. Important if several service networks across different platforms are to be interconnected. |
| Sensor Access Service | Basic interface for accessing sensor data, configuring a sensor and publishing sensor data. |
| Service Monitoring Service | Provides an overview about service instances currently registered within service network, e.g.  
1. Actual status (e.g. running, stopped, offline)  
2. Statistical information (e.g. average availability, response times) |
| User Management Service | Creates and maintains subjects (users or software components) including groups (of principals) as a special kind of subjects. |

Table 1: ORCHESTRA Architecture Services
• A specification of important feature types (e.g. document types) that may be re-used and refined in information models.
• A specification of a series of generic interface and service types that may (and should) be re-used by service modellers in the design of their geospatial SOA: starting from the interface types as the re-usable specification unit, assembling them to service types and possibly enriching them by domain-specific functionality.

Table 1 lists and briefly describes the currently specified architecture service types.

3.2 Concrete Service Platform

The ORCHESTRA Meta-model also provides rules describing how to map the abstract specifications to a concrete service platform. There are software tools available from ORCHESTRA that support this mapping process for the ORCHESTRA Web services platform. In this mapping process, UML information models have to be translated to XML/GML whereas UML interface and service models have to be mapped to WSDL documents.

For the service types listed in Table 1, ORCHESTRA also provides corresponding implementation specifications and implementations, most of them integrated in the common ORCHESTRA Service Container Framework and offered under an open source license.

When following a pure top-down approach with a 1:1 mapping from the abstract to the concrete service platform, the resulting Web services may not be compliant with existing OGC specifications. The ORCHESTRA answer to this problem is many-fold:

1) Those ORCHESTRA services that are concerned here because of their functional overlap with OGC services (e.g. Feature Access Service vs. OGC Web Feature Service, Map & Diagram Service vs. OGC Web Map Service, Catalogue Service vs. OGC Catalogue Service) may be (and have mostly been) implemented on top of existing OGC-compliant services in order to allow re-use of existing investments.

2) ORCHESTRA aims at providing OGC-compliant facades for its service implementations.

3) In the context of semantic extensions of the ORCHESTRA Architecture, how the mapping process may be made more flexible such that the discovery and invocation of OGC-compliant service operations may be mediated on the basis of the ORCHESTRA service specifications is the subject of on-going investigation.

Outlook

The focus of the ORCHESTRA Architecture as specified today lies on syntactic interoperability. The thorough analysis of user requirements has led to the specification of a series of generic services that provide powerful and indispensable functionality for the design of geospatial SOAs in the domain of environmental risk management and beyond. As concrete service platforms, W3C Web services and GML have been chosen as the current mainstream technology. Implementation specifications and implementations are available and will be offered under an open source license, too.
However, the ORCHESTRA Architecture has already opened the door towards enhancements. More powerful service platforms currently being specified by the Semantic Web Services community are emerging (OASIS 2006). Ongoing activities in the ORCHESTRA architecture group aim at extending the ORCHESTRA Reference Model such that these new technologies may be embedded and exploited. The application of semantic ORCHESTRA services in pilot test beds as, for example, presented in (Bügel/Hilbring 2007) are first steps in this direction and will provide important feedback about how semantics may be used in real world use cases.

4 Acknowledgements

This work has been co-funded by the European Commission's IST programme, under Unit G5 “ICT for the Environment”, through the Integrated Project FP6-IST 511678 ORCHESTRA (see http://www.eu-orchestra.org). Special thanks to Steven Frysinger for a thorough review of this article and to the members of the ORCHESTRA sub-project 3 "Open Architecture"

5 References


